AD-A107 917 BDM CORP MCLEAN VA F/G 15/3
MODULAR AIR DEFENSE EFFECTIVENESS MODEL, PROGRAM DOCUMENTATION --ETC(U)
JAN 80 M FILTEAU, B MACALEER, J T HAWKINS DNA001-79-C-0230
UNCLASSIFIED BDM/W-79-646-TR-VOL-3 DNA-5594F-3 NL

1 OF 1
AD
A107917

END
DATE
FILMED
1-82
DTIC

DNA 5594F-3

MODULAR AIR DEFENSE EFFECTIVENESS MODEL, PROGRAM DOCUMENTATION AND USER'S GUIDE

Volume III—MADEM Enhancement Specifications

The BDM Corporation
7915 Jones Branch Drive
McLean, Virginia 22102

31 January 1980

Final Report for Period 1 March 1979—31 January 1980

CONTRACT No. DNA 001-79-C-0230

**APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.**

THIS WORK SPONSORED BY THE DEFENSE NUCLEAR AGENCY
UNDER RDT&E RMSS CODE B325079464 V99QAXNI06215 H2590D.

Prepared for
Director
DEFENSE NUCLEAR AGENCY
Washington, D. C. 20305

Destroy this report when it is no longer needed. Do not return to sender.

PLEASE NOTIFY THE DEFENSE NUCLEAR AGENCY, ATTN: STTI, WASHINGTON, D.C. 20305, IF YOUR ADDRESS IS INCORRECT, IF YOU WISH TO BE DELETED FROM THE DISTRIBUTION LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

12 95

REPORT DOCUMENTATION PAGE		**READ INSTRUCTIONS BEFORE COMPLETING FORM**
1. REPORT NUMBER DNA 5594F-3	2. GOVT ACCESSION NO. AD A109919	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MODULAR AIR DEFENSE EFFECTIVENESS MODEL, PROGRAM DOCUMENTATION AND USER'S GUIDE Volume III—MADEM Enhancement Specifications		5. TYPE OF REPORT & PERIOD COVERED Final Report for Period 1 Mar 79—31 Jan 80
		6. PERFORMING ORG. REPORT NUMBER BDM/W-79-646-TR
7. AUTHOR(s) M. Filteau J. T. Hawkins L. Elfes B. Macaleer J. Philips		8. CONTRACT OR GRANT NUMBER(s) DNA 001-79-C-0230
9. PERFORMING ORGANIZATION NAME AND ADDRESS The BDM Corporation 7915 Jones Branch Drive McLean, Virginia 22102		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Subtask V99QAXNI062-15
11. CONTROLLING OFFICE NAME AND ADDRESS Director Defense Nuclear Agency Washington, D.C. 20305		12. REPORT DATE 31 January 1980
		13. NUMBER OF PAGES 96
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This work sponsored by the Defense Nuclear Agency under RDT&E RMSS Code B325079464 V99QAXNI06215 H2590D.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Modeling
Simulation
Air Defense Models
Discrete Event Simulation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document is one of three volumes which provide documentation and user guidelines for the Modular Air Defense Effectiveness Model (MADEM). The three volumes are: VOLUME I—MADEM Analyst Manual; VOLUME II—MADEM Programmer Manual; and VOLUME III—MADEM Enhancement Specifications.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

This document is a companion to the Modular Air Defense Effectiveness Model (MADEM) Analyst and Programmer manuals. It describes a series of enhancement designs completed during the MADEM Final Development Project. These designs have not been implemented in the current version of MADEM and have been documented for discussion purposes only.

Because of the complex nature of the MADEM simulation and its relationship to these enhancement designs, the user is advised to read the MADEM Analyst Manual before reading this document.

AD A107915

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

LIST OF ILLUSTRATIONS

SECTION I
INTRODUCTION

The purpose of this addendum is to document the enhancement designs for the MADEM final development project. Chapter II provides descriptions of each enhancement along with a brief summary of the software modifications required to implement the enhancements. In some cases the designs were sufficiently complex to require development of extensive program design language (PDL) specifications. These program design language specifications are listed in the Appendices.

These enhancements include:

(1) Treatment of communications between CRC's. Incorporation of this effect will portray CRC operations intended to prevent multiple engagements of the same target be different CRC's;

(2) Implementation of an echelon of command above the CRC. This simulated AAFCE commander will preceive the overall threat and mass air and ground based assets to meet it. In particular, this feature will capture some of the major dynamic factors of the defensive counter-air battle;

(3) Treatment of beyond-visual-range (BVR) air-to-air engagements and additional interceptor tactics including representation of tankers, and treatment of disengagement logic for interceptors. This feature will improve the realism of the air-air portion of the model;

(4) Representation of mobile surface-to-air missile (SAM) fire units. This enhancement will provide additional analytic resolution into issues of the survivability and capability of developmental SAM systems to be deployed in the near future;

(5) Modification of existing modeling of short-range air defense systems (SHORADS) to include treatment of SHORAD attrition as a function of expected unit densities. This feature will improve the realism of the model and provide for user flexibility in modeling SHORADS effects and trade-offs; and,

(6) Modification of the existing threat planner module to allow for representation of alternate threat characteristics such as coordinated surface-surface missile and air strikes on NATO.

None of these enhancements have been implemented in the current version of MADEM. Accordingly, these designs should be viewed only as documentation of design work completed to date. They may therefore be modified to meet user requirements with minimal difficulty.

SECTION II
ENHANCEMENT SPECIFICATIONS

A. COMMUNICATION BETWEEN CRC's

1. Description

The purpose of this enhancement is to prevent multiple engagement of the same target by different CRC's. Communication of assignment information among CRC's will be simulated through the addition of an assignment flag to Red flight status data structures. This new flag, which tags the Red flight as "assigned" or "not assigned", will be accessed by Blue CRC's that perceive the Red flight. If the Red flight has already been assigned by another CRC, the CRC currently considering it for assignment will ignore it and proceed with consideration of other Red flights.

The identification of a Red flight as "assigned" or "not assigned" will be handled probabalistically. User input probability levels will be placed on the true recognition of the assignment, and the misidentification by IFF. With this enhancement in place the user can, in effect, specify the degree of IFF error and of multiple assignments on a red target.

In addition to the simulation of CRC communications, this enhancement will also include new CRC engagement control options. Currently, CRC's attempt to assign all perceived unassigned enemy aircraft to airborne interceptor flights. If interceptors are unavailable, the enemy aircraft are assigned to an available BOC. This enhancement will allow the user to specify a priority for first assignment attempts by CRC's to either interceptors or BOC's. This priority will be based on a random draw against a weighting factor input by the user.

2. Software Modifications

The enhancements described above will require modification of twelve (12) subroutines, addition of a new common block and modification of one MIDAS data structure.

The data input subroutine OTHERDAT will be modified to allow the user to input five new variables. These new variables, which will be placed in a new common block for use by the assign module, are defined below:

ARL1 - probability of CRC recognizing that an assigned target has been assigned.
Range (0.0 to 1.0) Real

ARL2 - Probability of CRC recognizing that an unassigned target is unassigned.
Range (0.0 to 1.0) Real

FFLVL1 - Probability of correct IFF on friendly aircraft by CBC.
Range (0.0 to 1.0) Real

FFLVL2 - Probability of incorrect IFF on enemy aircraft by CRC.
Range (0.0 to 1.0) Real

APRTY - Range (0.0 - 1.0) sets weighting factor on whether interceptors or battalion get first attempt in assignments random draw done on a target by target basis.

In addition a new field "ASGNFLG" will be added to the MIDAS data block ARCFTSTATUS. This field will hold a bit flag to mark a Red aircraft as "assigned" or "not assigned". A (Ø) in this field indicates that the flight is not assigned. This flag may be set using the following pointer sequence:

$ PTGTSB.PARCFTSTAT.ASGNFLG$ = 1 To Set
$ PTGTSB.PARCFTSTAT.ASGNFLGS = Ø To Reset

This bit flag must be set or reset in the following routines:

BADMOVE
BTNASIN
CRCDIES
CRCKIL
CRCLOSS
INTASIN
CRC2INT
INT2CRC
NEWMOVE

The CRC assignment subroutine ASSIGN and the two subroutines (BTNASIN and INTASIN) it calls to allocate either BOC's or interceptors must be modified to include the logic changes inherent in this enhancement. The changes to BTNASIN and INTASIN will be relatively simple and do not require a major rewrite of their PDL (program design language). However, the modifications ASSIGN require to implement these enhancements are substantial and neccessitate a complete rewrite of the PDL for this subroutine. The revised PDL for the ASSIGN *subroutine* is listed in Appendix A.

B. OVERALL THREAT PERCEPTION AND RESOURCE ALLOCATIONS

1. Description

This enhancement is designed to simulate the echelons of command above the CRC. The simulation of the SOC and ATAF command levels will perceive the overall threat and mass air assets to meet it. In particular, this feature will capture some of the major dynamic factors of the defensive counter-air battle.

Threat perception is based upon the ability of CRC's to assign interceptors to engage Red penetrators. Perception of threat and reallocation of interceptors among CRC's are triggered by a REALLOCATE event. Reallocate events occur at fixed time intervals set by the user or when a CRC exceeds its user defined overload threshold. The purpose of this dual triggering mechanism is to allow the user to tailor the sensitivity of the threat perception and reallocation process to most different doctrinal concepts. It also provides a simple mechansim for fine-tuning the system to prevent either over or under-reaction to threats. Moreover, the dual triggering system allows the user to design an ATAF which is sensitive to long term situations through the fixed interval trigger, and sensitive to short-term threats on a given CRC through the overload trigger.

The perception/reallocation process is driven by a categorization of CRC's as sources of supply or demand for aircraft resources. CRC's that could not assign either BOC's or FLIGHTS to engage red penetrators during the preceding time interval are considered to be demand CRC's. Demand is measured in terms of the number of flights that would have been required to intercept the unassigned Red flights. CRC's that had aircraft left over after assigning all incoming Red flights are considered to be supply CRC's. Aircraft on air bases tasked by supply CRC's may be reallocated to air bases tasked by demand CRC's.

Aircraft flights are reallocated to demand CRC air bases from supply CRC air bases using the following criteria:

- DEMAND CRC CRITERIA
  (1) Demand Level
  (2) Airbase Damage Levels
  (3) Service Capacity of Air Bases for Various Aircraft Types

- SUPPLY CRC CRITERIA
  (1) Supply Level or User Specified Priority
  (2) Availability of Various Aircraft Types on Air Bases
  (3) Distance of Supply Air Bases From Demand Air Bases
  (4) Minimum Aircraft Stockage Levels for each Air Base
  (5) Maximum Percentage of Aircraft that can be Reallocated in a Single Event.

The priority assigned to the satisfaction of demand CRC needs is primarily a function of the CRC's demand level. However, the actual number and type of flights allocated to its air bases will depend upon their damage levels and their ability to service various types of aircraft. Aircraft reassignment priority is given to the least damaged air bases on the assumption that their servicing and launching capabilities would be greater. Similarly, aircraft are assigned only to airbases which have the capacity to service additional aircraft of the specified type. Aircraft type service capacities for each air base are input by the user.

Aircraft flights are allocated from supply CRC's on the basis of a supply priority list. This priority list may be constructed in two ways. It may be ordered with the CRC's having the greatest supplies at the top or (at the user's option); the list may be ordered on the basis of user specified supply priorities for each CRC. Regardless of priority, aircraft will not be taken from CRC's unless they have a surplus of aircraft of the type required to meet a specific demand CRC's needs. The reallocation algorithm is designed to minimize the distance reallocated flights must travel and to insure that user specified minimum aircraft stockage levels at each air base are met. In addition the user may specify the maximum percentage of aircraft on hand that can be reallocated from an air bases in a single reallocation event. These constraints are provided to allow the user better control of the rate and level of resource depletion at supply air bases. It should be noted that the determination of supply and demand CRC's is not fixed from reallocation event to reallocation event. This allows the ATAF to adjust to shifting threats.

The specific logic for the threat perception and resource reallocation algorithm is outlined in the program design language for this enhancement in Appendix B. This logic has been tested manually on simple example cases like the one illustrated in Figure 1. The base conditions and resulting reallocation of interceptors for this case are shown in Figure 2.

In Figure 1 CRC's 1 and 2 are able to assign incoming threats with no surplus while CRC's 2 and 4 have surplus assets of 5 and 12 flights, respectively. CRC 5 has experienced an overload which resulted in a demand of 15 flights. The availability of specific aircraft types is shown in the base condition tables in Figure 2. The base condition table for the demand CRC also indicates the damage level of each air base in the upper right hand corner of its air base subtable. Following the algorithm (ATAF INTERCEPTOR REASSIGNMENT) generates the reallocation table shown for CRC 5 in Figure 2. These results are graphically displayed in Figures 3, 4, and 5.

A primary assumption of the algorithm described above and in Appendix B is that unassigned Red flights in the preceding time interval is an adequate measure of current threat for use in massing resources. The major limitation of this approach is that it is purely reactive. Depending upon the reallocation time interval it is conceivable that reallocated flights would not become useful to the demand CRC until after the actual threat has passed. Moreover, it may lag actual demands on CRC resources to

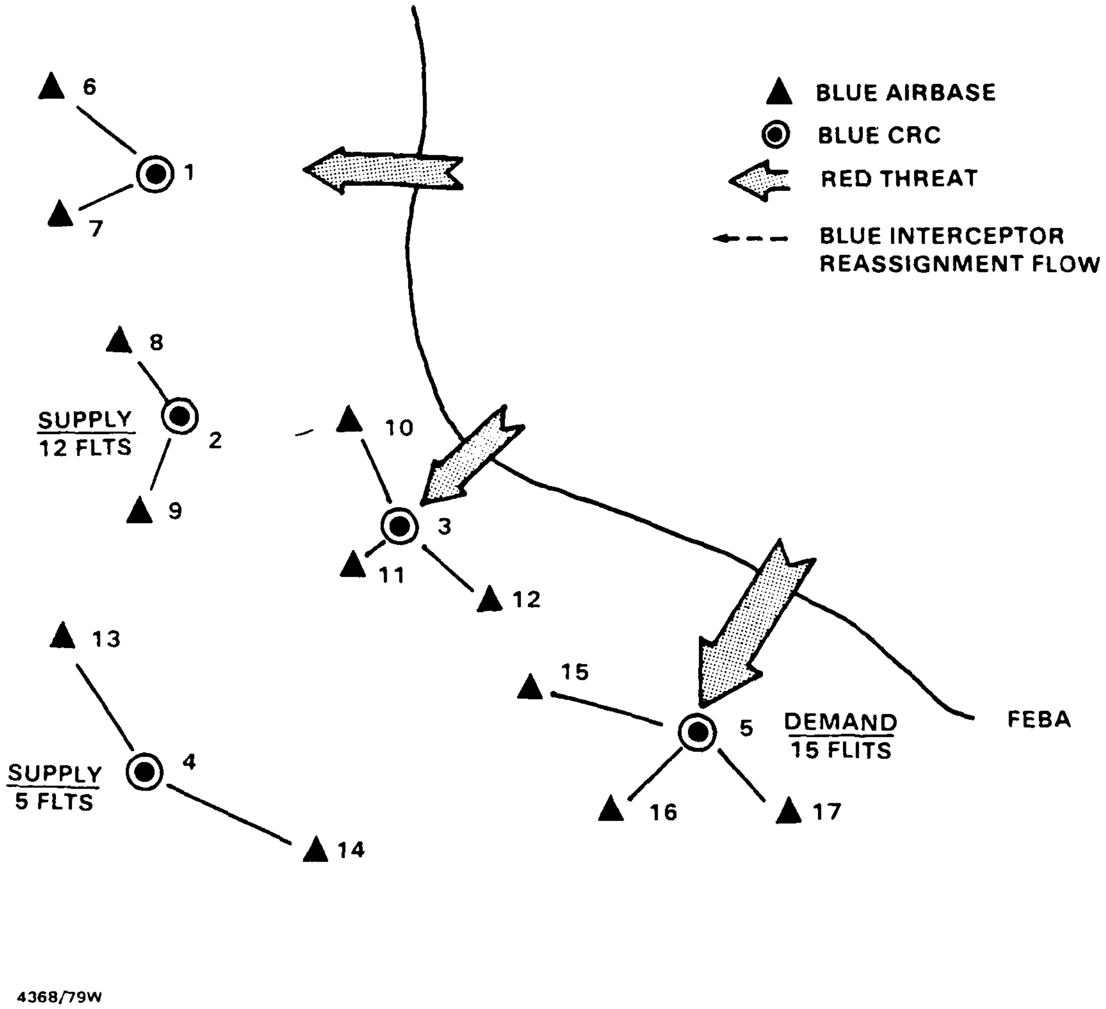
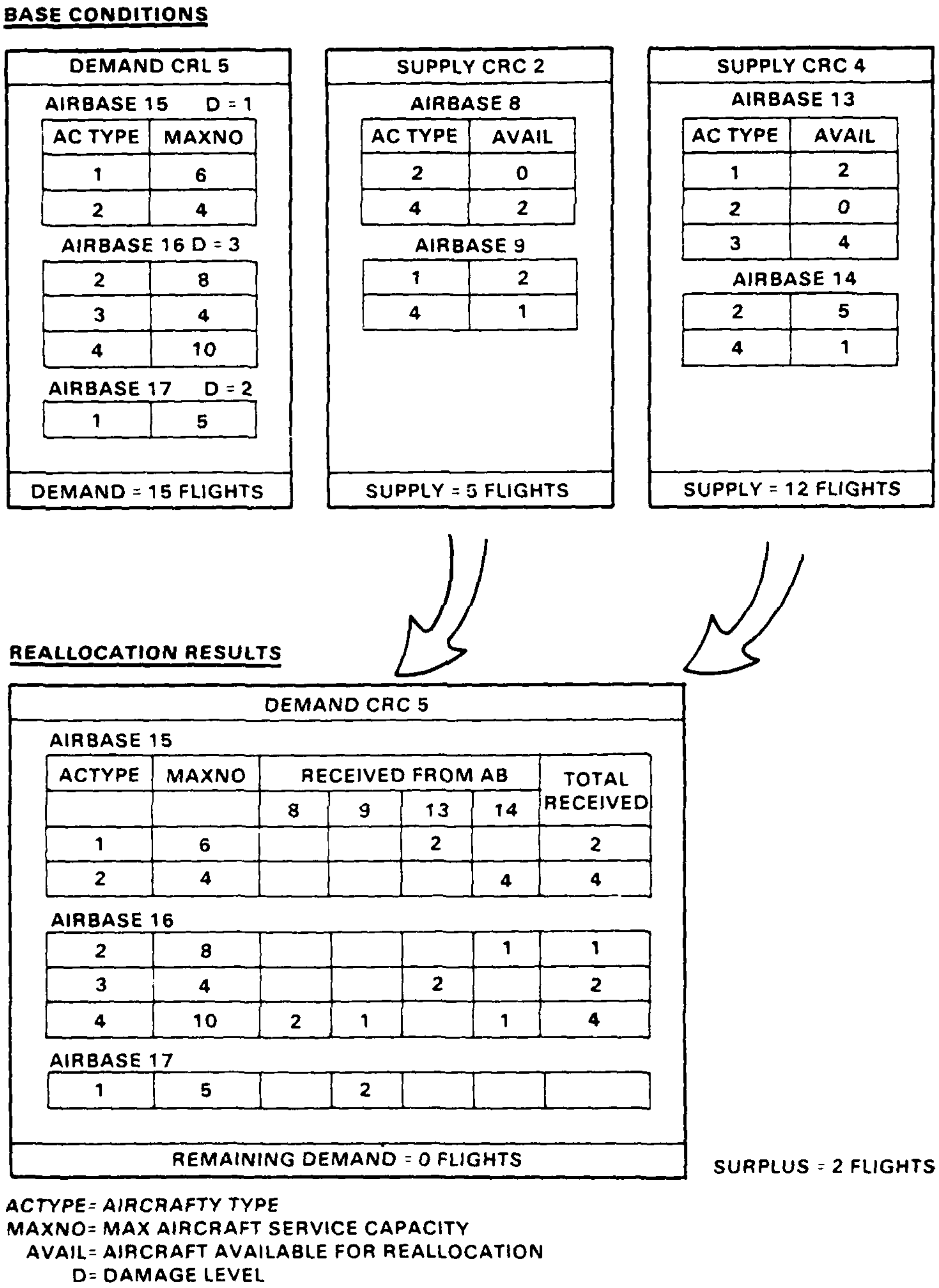


Figure 1. Interceptor Reallocation Example Case



BASE CONDITIONS

DEMAND CRL 5	
AIRBASE 15	D = 1
AC TYPE	MAXNO
1	6
2	4
AIRBASE 16	D = 3
2	8
3	4
4	10
AIRBASE 17	D = 2
1	5
DEMAND = 15 FLIGHTS	

SUPPLY CRC 2	
AIRBASE 8	
AC TYPE	AVAIL
2	0
4	2
AIRBASE 9	
1	2
4	1
SUPPLY = 5 FLIGHTS	

SUPPLY CRC 4	
AIRBASE 13	
AC TYPE	AVAIL
1	2
2	0
3	4
AIRBASE 14	
2	5
4	1
SUPPLY = 12 FLIGHTS	

REALLOCATION RESULTS

DEMAND CRC 5

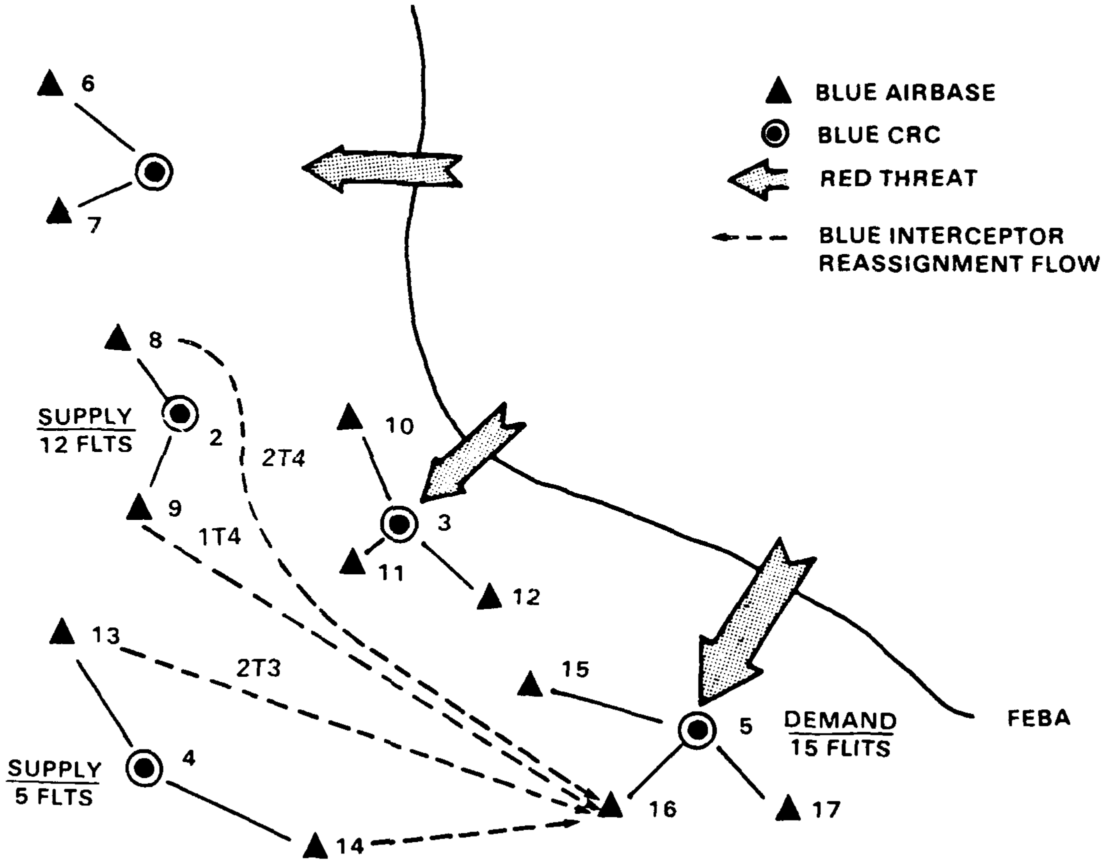
ACTYPE	MAXNO	RECEIVED FROM AB				TOTAL RECEIVED
		8	9	13	14	
AIRBASE 15						
1	6			2		2
2	4				4	4
AIRBASE 16						
2	8				1	1
3	4			2		2
4	10	2	1		1	4
AIRBASE 17						
1	5		2			

REMAINING DEMAND = 0 FLIGHTS

SURPLUS = 2 FLIGHTS

ACTYPE= AIRCRAFTY TYPE
MAXNO= MAX AIRCRAFT SERVICE CAPACITY
AVAIL= AIRCRAFT AVAILABLE FOR REALLOCATION
D= DAMAGE LEVEL

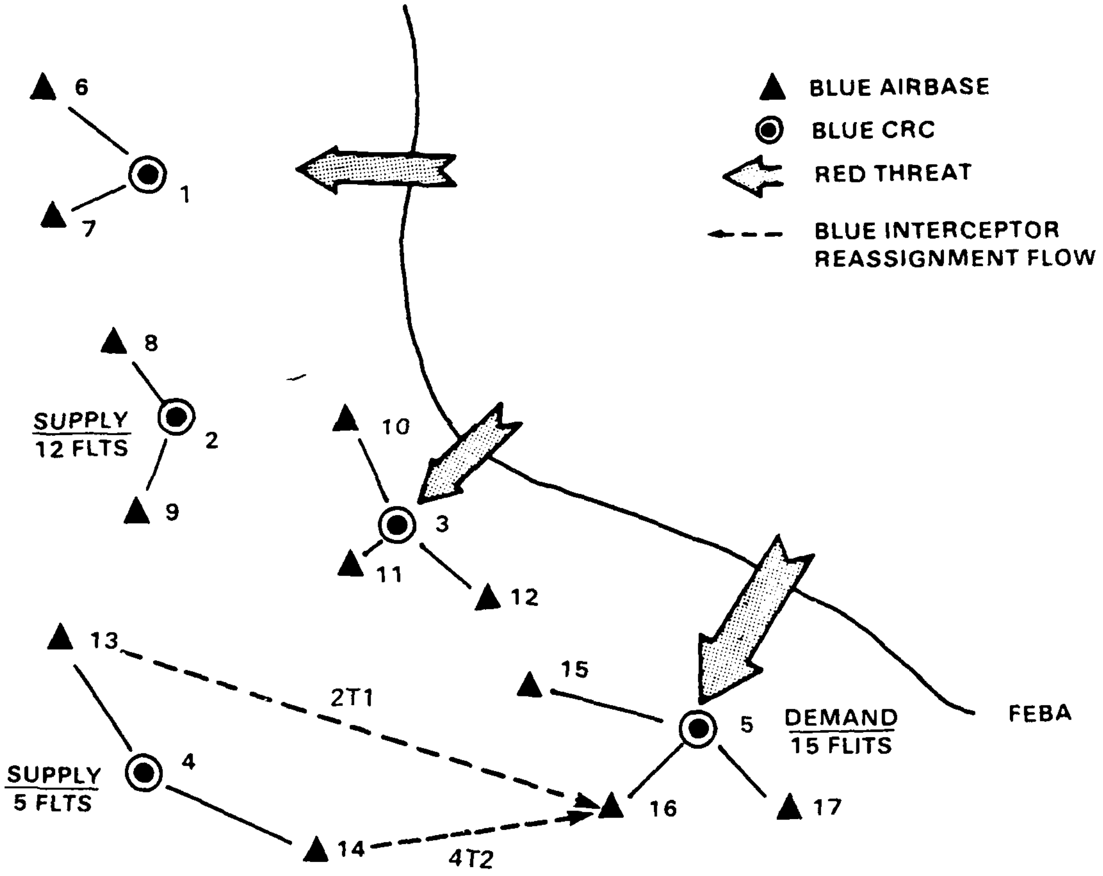
4368/79W Figure 2. Interceptor Flight Reallocation Example Data



NOTE: 2T4 = 2 TYPE 4 AIRCRAFT FLIGHTS

4368/79W

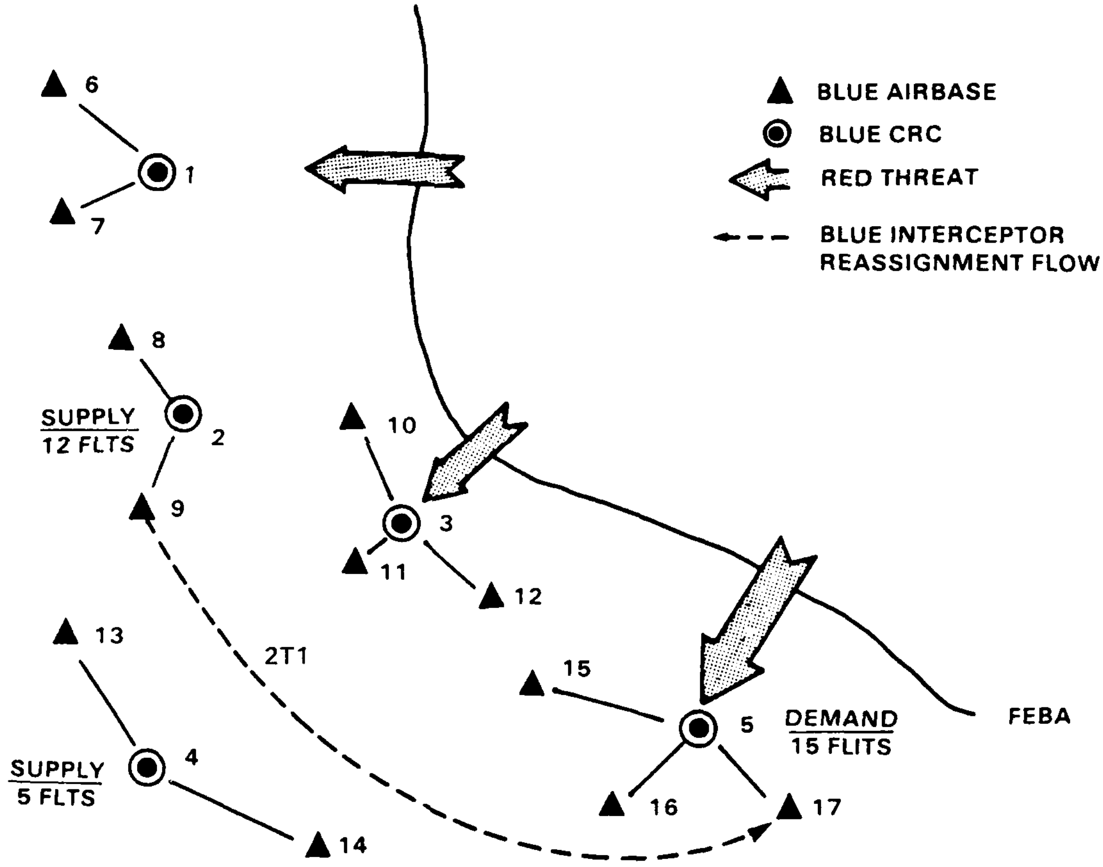
Figure 3. Reallocation Of Interceptor Flights To Airbase 16



NOTE: 2T1 = 2 TYPE 1 AIRCRAFT FLIGHTS

4368/79W

Figure 4. Reallocation Of Interceptor Flights To Airbase 15



NOTE: 2T1 = 2 TYPE 1 AIRCRAFT FLIGHTS

4368/79W

Figure 5. Reallocation Of Interceptor Flights To Airbase 17

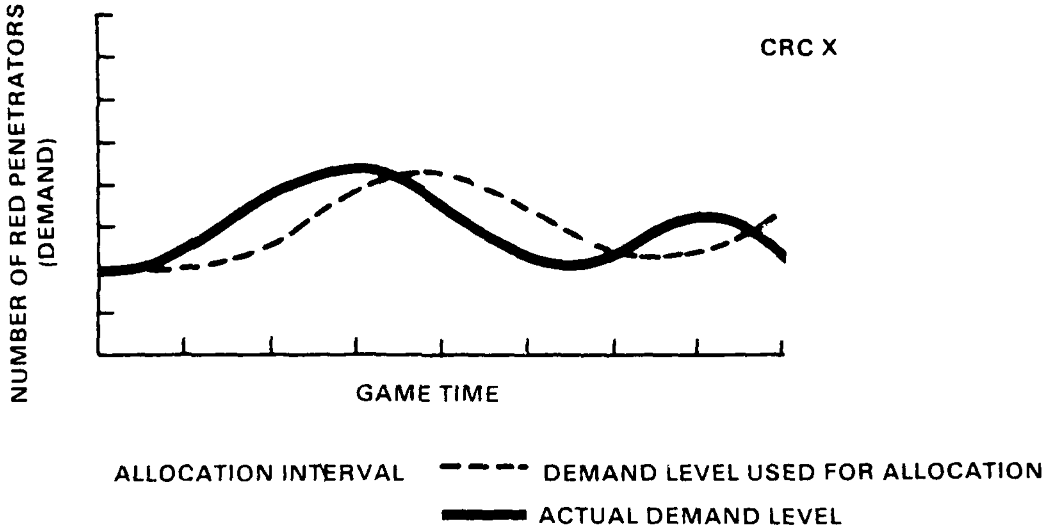
the point where its utility in "massing" resources to meet perceived threats may be limited. This problem is illustrated in Figure 6.

In Figure 6, the reactive approach to interceptor reallocation leads to a consistent under allocation of resources in the crucial early stages of mounting threat to CRC X. This problem can be dealt with by using a forecasted demand approach to reallocation. This approach attempts to anticipate demands upon CRC's and allocate interceptors with enough lead time to mass to "meet" the threat rather than react to it. The mechanics of this approach, which is based upon time series analysis of incoming penetrators at each CRC, is illustrated in Figure 7. In Figure 7 the ATAF has calculated a trend projection of the incoming penetrator curve from T2 to T3. This results in a predicted demand for 6 interceptors for T3. Consequently the demand allocated for in T2 will be the anticipated 6 penetrators rather than the 4 actually observed at T2.

The conceptual strength of this forecasted demand approach is illustrated in Figure 8. The situation here contrasts sharply with Figure 6 in that major increases in threat level are anticipated rather than lagged behind as a result reallocations can be made in advance of the arrival of penetrator flights. It is felt that this approach in combination with the reallocation algorithm and its associated dampening parameters will lead to a "rational" reallocation of interceptor resources.

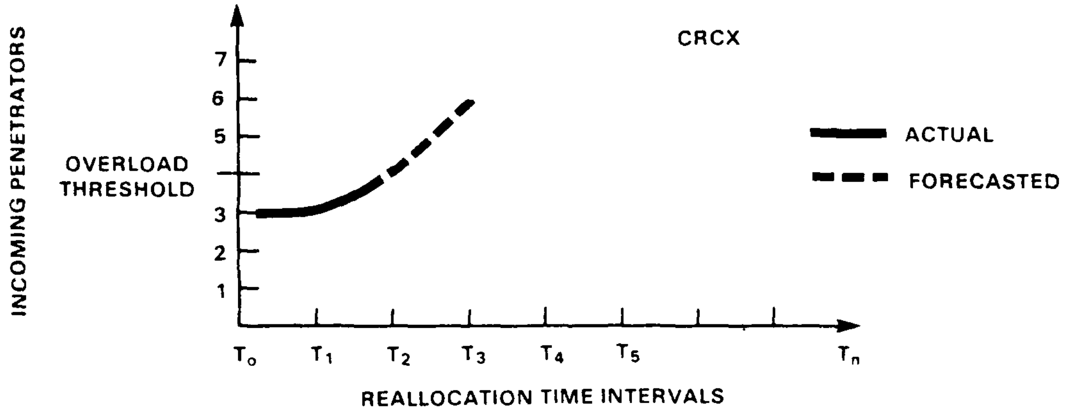
2. Software Modifications

This enhancement will require modification of the existing data input routine OTHERDAT to allow input of the new parameters and the SELECT routine to allow processing of the new REALLOCATE software module. This new module will consist of at least seven new subroutines designed to implement the algorithm discussed in Section 1. The program design language for these routines is listed in Appendix B. REALLOCATE module subroutines include:



4368/79W

Figure 6. Reactive Approach

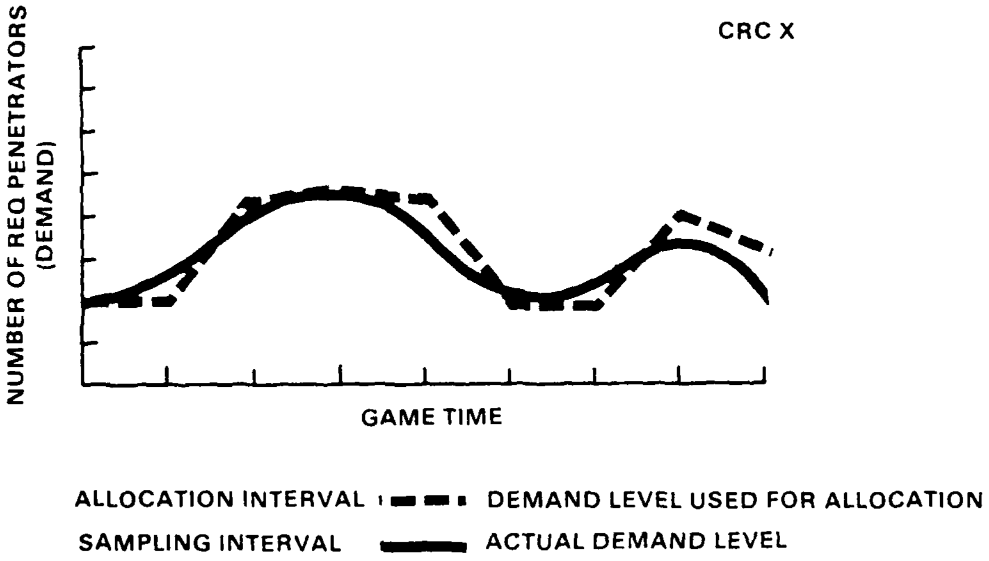


EXPLANATION:

OVERLOAD THRESHOLD WAS REACHED AT T_2. A TREND PROJECTION OF THE INCOMING PENETRATORS FROM T_2 TO T_3 RESULTS IN A DEMAND OF 6 FOR T_3. CONSEQUENTLY THE DEMAND ALLOCATED FOR IN T_2 WILL BE 6 FLIGHTS RATHER THAN THE 4 OBSERVED AT T_2.

4368/79W

Figure 7. Forecasting Demand for Interceptor Flights by a CRC Using the Time Series Analysis Approach



4368/79W

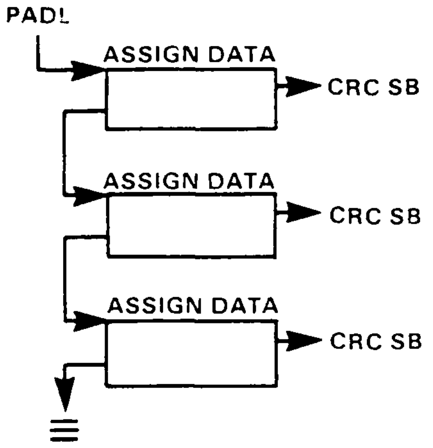
Figure 8. Forecasted Demand Approach

ATAF	-	Interceptor Reassignment Control Routine
SDSORT	-	Creates Sorted Demand and Supply CRC Lists
DSTSRT	-	Sorts Supply Air Bases by Distance from Demand Air Bases
DAMSRT	-	Sorts Demand Air Bases by Damage Level
CADR	-	Assignment Distribution Recorder
VCSCL	-	Updates Assignment Distribution List
TFURK	-	Forecasts Incoming Penetrators For Each CRC (Method Under Investigation)

In addition to these software modifications several new data structures will be created. These structures and their contents are specified in Figures 9, 10, and 11.

The CRC assignment distribution list shown (Figure 9) is used to keep track of the penetrator level at each CRC and the assets assigned to meet the threat by the CRC.

The supply and demand CRC lists (Figure 10) are used to temporarily sort CRC's by supply and demand levels. They are the basic mechanism through which the reallocation is carried out. Similarly, the supply and demand air base lists (Figure 11) are used to sort supply and demand air bases.



ASSIGN DATA

PNEXT	PABLST
PLRCSB	
DEMAND	PRIOR
IASSGN	IAVAIL
BASSGN	BAVAIL
OTHRES	

PNEXT — POINTER TO NEXT ASSIGN DATA BLOCK IN THE LIST

PABLST — POINTER TO CRC's AIRBASE LIST

PCRCSB — POINTER TO THE CRC's SB BLOCK. SET NEGATIVE IF THE CRC IS DEAD

DEMAND — NUMBER OF RED FLIGHTS WHICH COULD NOT BE ASSIGNED AND WHICH WERE NOT ASSIGNED BY OTHER CRC's

PRIOR — SUPPLY PRIORITY FOR CRC SET, BY THE USER

IASSIGN — NUMBER OF INTERCEPTOR ASSIGNMENTS MADE

IAVAIL — NUMBER OF INTERCEPTORS AVAILABLE (SUPPLY)

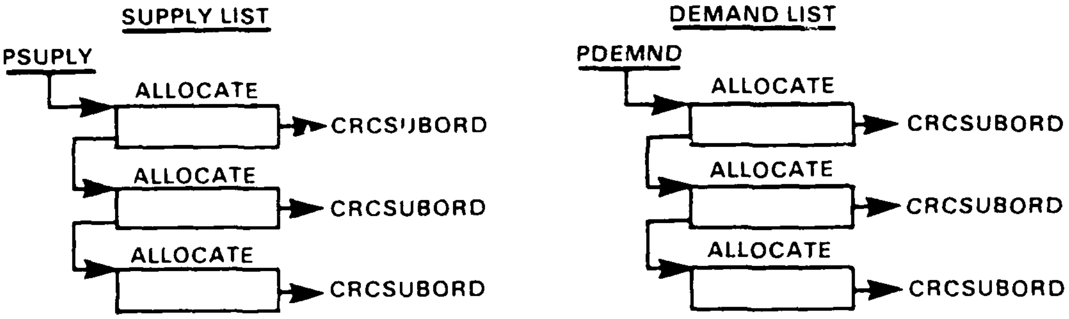
BASSIGN — NUMBER OF BOC ASSIGNMENTS MADE

BAVAL — NUMBER OF BOC's AVAILABLE

OTHRES — OVERLOAD THRESHOLD FOR CRC SET BY THE USER. WHEN INCOMING TRACKS EXCEED THIS LEVEL, A REASSIGNMENT EVENT IS SCHEDULED FOR IMMEDIATE OCCURRENCE.

4368/79W

Figure 9. CRC Assignment Distribution List (Permanent)



ALLOCATE

PNEXT	PABLST
DEMAND	PRIOR
SUPPLY	OTHERS

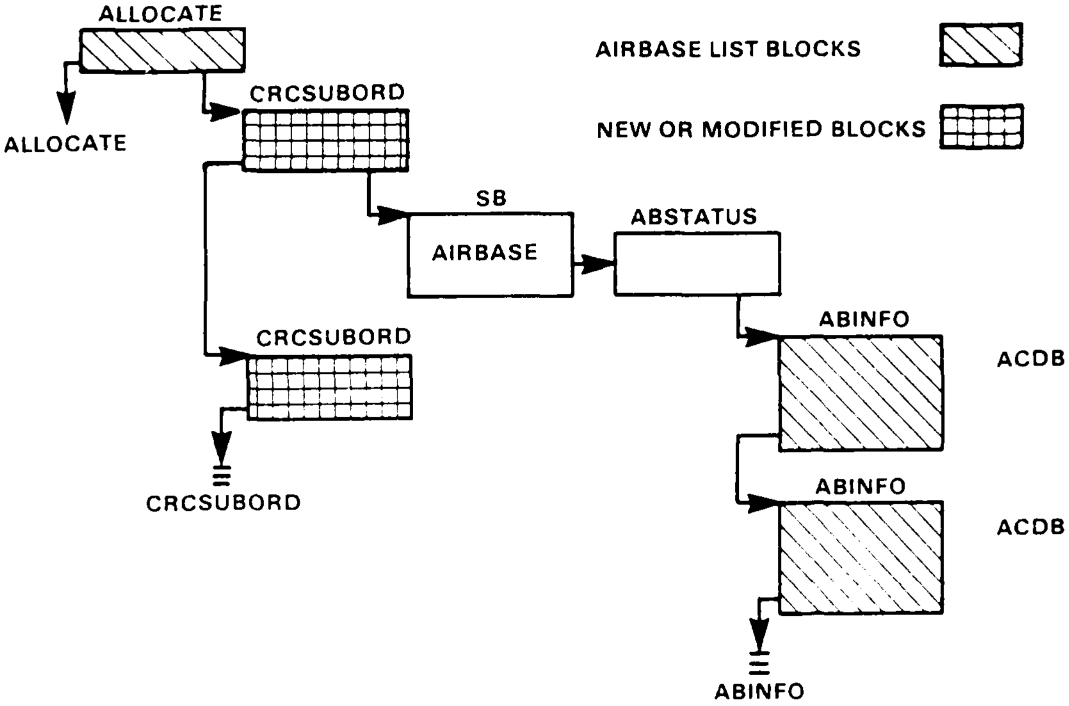
PNEXT — POINTER TO NEXT ALLOCATE BLOCK IN THE LIST
PABLST — POINTER TO CRC's AIRBASE LIST
DEMAND — NUMBER OF UNASSIGNED RED FLIGHTS
PRIOR — SUPPLY PRIORITY
SUPPLY — AVAILABLE AIRCRAFT AT BASES COMMANDED BY THE CRC (SUPPLY) SUPPLY = IAVAIL IN THE ASSIGNED DATA BLOCK
OTHRES — OVERLOAD THRESHOLD

SUPPLY LIST — SORTED BY PRIOR LEVEL OR SUPPLY LEVEL HIGHEST SUPPLY OR PRIORITY CRC AT TOP

DEMAND LIST — SORTED BY DEMAND LEVEL. HIGHEST DEMAND AT TOP

4368/79W

Figure 10. Supply And Demand CRC Lists (Temporary)



NEW FIELDS IN ABINFO BLOCK

MINNO — MINIMUM NUMBER OF AIRCRAFT OF THIS TYPE ALLOWABLE ON THIS BASE. INPUT BY USER.

MAXNO — MAXIMUM NUMBER OF AIRCRAFT OF THIS TYPE THAT CAN BE SERVICED ON THIS BASE. INPUT BY USER.

MAXRAL — MAXIMUM PERCENTAGE OF AIRCRAFT ON HAND (NOACOH) THAT CAN BE REALLOCATED IN A SINGLE REALLOCATION EVENT. INPUT BY USER.

DEMAND AIRBASE LIST — SORTED BY DAMAGE LEVEL IN THE ABSTATUS BLOCK. LEAST DAMAGED BASES AT TOP.

SUPPLY AIRBASE LIST — SORTED BY DISTANCE FROM DEMAND AIRBASE CURRENTLY BEING CONSIDERED.

4368/79W

Figure 11. Supply And Demand Airbase Lists

C. INTERCEPTOR OPERATIONS

Interceptor operations enhancement includes three distinct enhancements:

(1) treatment of beyond-visual-range (BVR) air-to-air engagements,

(2) representation of tankers and their use by interceptors, and

(3) treatment of disengagement logic for interceptors.

These features are intended to improve the realism of air-to-air engagements. Each of these three enhancements are discussed separately in this section.

1. Tankers

a. Description

The purpose of this enhancement is to allow tanker refueling of Blue aircraft. Aircraft capable of tanker refueling will have the option of returning to the airbase or seeking a tanker to refuel. Tankers themselves will be treated as aircraft with no defense capabilities, other than evasive maneuvers. The user will input "anchor" hexes in which the tankers orbit while refueling other aircraft.

The locations of anchor hexes, the number of tankers to be constantly kept at each anchor, the air base responsible for each anchor hex, the total number of tankers at each air base, and various data on tankers will all be input by the user. When a Blue CRC sends scramble messages, the tankers will begin to take off. The simulation will assign tankers to each anchor hex, and schedule their take off times in such a way that if each tanker was constantly refueling then the anchors would always have the appropriate number of tankers. The actual number of tankers in the hex is a function of tanker demand, the number of tankers initially available, and the attrition of tankers during the simulation.

Once the tankers have begun taking off, they fly directly to their anchor hex. Once in the anchor hex, the tanker will orbit and refuel until either (1) its fuel reserve is depleted or (2) after a set period of time the tanker is not busy and its replacement has safely arrived. The tanker then returns to the air base for refueling and subsequent takeoff.

The tanker may be attacked and destroyed at any time during its flight. If this happens, it will be replaced only if additional tankers are available.

When a flight decides to refuel at a tanker, it flies directly to the nearest anchor hex with available fuel. If none are available, or if the only tankers available are too far way, then the flight returns home. When a flight arrives at an anchor hex, the flight may have to orbit until a tanker is available. Once the flight is refueled, the flight goes back into orbit, notifies the CRC of its status, and awaits orders from the CRC. At any time during this refueling process, however, the flight may divert from its plans to attack or defend itself from a Red aircraft. The tanker must depend on its interceptors in the area to defend the tanker from Red attack.

This enhancement is relatively complex, as indicated by the software modifications section. If it is desired to scale down the complexity of this enhancement, there is an option that is less realistic but simpler to implement. All tanker movements could be eliminated by merely continuous presence at a given hex location until a tanker was shot down. This design would produce, however, the unrealistic effect of having a limitless supply of fuel in the air. Also, when one of these tankers was shot down, it would not have a replacement scheduled to arrive shortly, as is the case in the more complex design. To use the simpler design, ignore any modifications that have to do with tanker movement, and place the tankers permanently in the appropriate anchor hex.

In both the simple and complex cases, tankers may be implemented for Reds by mirroring the Blue changes in the Red code. The difference would be that the PLAN module would schedule take off and placement of Red tankers.

b. Software Modifications

The enhancement described above will require the following software modifications:

(1) Additional user input

(2) the modification of eight existing subroutines:

BLUTKOF	Blue take off
CRCKIL	CRC ponders a kill
BLULAND	Blue aircraft land
FLITE	determine next fly maneuver
COMMAND	process flight action codes
SEMANT	(preprocessor) semantic processing
FUELCHK	aircraft fuel check
OTHRDAT	data input processing

(3) the addition of six new subroutines

TNKINIT	Tanker-Anchor initial processing
TNKCALL	Tanker-Anchor calculations
CHZANCR	Choose a Tanker-Anchor to go to
FLTANCR	Flight arrives at Anchor
TANCHOR	Tanker-Anchor processing
FLTFINR	Flight finished refueling

(4) the modification of three data blocks

ACDB	Aircraft data base
PROFILEDBLOK	Flight profile
SB	Score board

(5) the addition of one new Tanker data structure, consisting of two old blocks and three new blocks:

TNKANCRSTATUS	Tanker-Anchor status board
TANKERLIST	Tanker list (for each anchor)
TANCHRQ	Anchor Queue

(6) the addition of one common variable:

PTANCHR	pointer to list of all anchor status boards

(7) the addition of three new events:

a) flight arrives at anchor hex

b) anchor processing

c) flight finished refueling from tanker

(8) the modification of one flight action code (code 4)

(9) the addition of flight action code:

12 - refueling from tanker

These modifications are described in detail below.

1) User Input Modifications

The following 13 types of input will be needed to describe the tankers and their operations:

INPUT 1. TANKABTIME - the time it takes a tanker to refuel and be ready for take off after landing at air base. (seconds)

INPUT 2. TANKFLTSPEED - tanker ground speed enroute to anchor hex or air base. (meters/sec)

INPUT 3. TANKORBSPEED - tanker ground speed while orbiting in anchor hex. (meters/sec)

INPUT 4. TANKFLG - flag indicating aircraft type is tanker. (-1=tanker)

INPUT 5. TANKREF - tanker refueling capability flag, indicates whether or not an aircraft can use tankers

INPUT 6. RELTIME - time it takes a particular type of aircraft to refuel using a tanker. (seconds)

INPUT 7 ALT2AB - altitude of tankers while traveling to or from air base.

INPUT 8. ALTANCR - altitude of tankers while orbiting in the anchor hex.

INPUT 9. NUMACRFT - number of tankers initially assigned to an air base.

INPUT 10. Air base to Anchor command and control relationships

INPUT 11. Locations of Anchors by hex number

INPUT 12. TNEEDED - Number of tankers needed at a given anchor hex

INPUT 13. The rank, or priority, by which anchors are allocated tankers. This input is implied by the order of inputs 10 and 11.

The first nine of these inputs will come from the DATFILE, the remaining inputs through UOIL sentences. Of the nine DATFILE inputs, only INPUTS 4, 5, and 6 are new INPUT fields. The other six inputs are additional definitions of existing fields, which are to be assumed when a tanker is being described.

DATFILE Input Modifications

There may be only one formation with tankers; this formation must have exactly one flight with exactly one aircraft (tanker) in the flight. There may be no class 6006 (uis'tion data base) for tanker flights. The following DATFILE field will be given additional definitions applying to tanker, or be added as new fields, as described below:

CLASS 6003 - *Aircraft Data Base*

CARD B

Field 2 - TANKABTIME
Field 3 - TANKFLTSPEED
Field 4 - TANKORBSPEED
Field 6 - Dummy vale = Ø (see BUR enhance)
Field 7 - TANKFLG, must be -1 for tankers (see BUR ennance)
Field 8 - Dummy value = Ø for tankers for non-tanker aircraft, this field is TANKREF (new)
Field 9 - For tankers, dummy value = Ø (new)
For other aircraft, this field is RELTIME

CLASS 6005

Field 2 - ALT2AB
Field 3 - ALTANCR

CLASS 6002

Fields all have same definition, but for tankers fields 3, 4, and 5 of card A must have a value of 1. Field 9 must have a zero value.

CLASS 6008

No change in definition, use to describe NUMACRFT.

UOIL Input Modifications

There will be one new sentence structure defined the UOIL, and two old sentences will be used to describe Anchors for tankers.

The examples below are self explanatory:

(1) 21 AIRBASE COMMANDS 5 ANCHOR

(2) 5 ANCHOR IS AT HEX 7777123

(3) 5 ANCHOR DESIRES 2 TANKERS (new)

The "RANK" of anchor importance, which affects tanker allocations only if there is a shortage of tankers, is implied by the order of entry. The first Anchor described will be the lowest priority.

2) *Subroutine Modifications*

Subroutines OTHRDAT and SEMANT will need to be modified to handle the new input specifications listed above. SEMANT will also need to call TNKINIT (which calls TNKCALC) to do initial anchor calculations and set up the anchor data base.

In the TWOER module, subroutine BLUTKOF must be modified to handle tanker take off and scheduling. When a scramble message is in effect, BLUTIKOF will launch one tanker for each anchor, and then schedule further tankers to fulfill the demand at each anchor.

Subroutine COMMAND will be modified to handle action code 4 - flight orbit in a hex - so that tankers can use this code to plan the anchor orbit pattern. Command action code 12 will be added for use as "flight arrives at anchor hex for refueling".

Subroutine FLITE will be modified to provide an event trace for tankers.

Subroutine CRCKIL, which handles flights shot down, will be modified to provide a tanker event trace as well as to

Subroutine CRCKIL, which handles flights shot down, will be modified to provide a tanker event trace as well as to try to schedule replacements for tankers shot down.

Subroutine BLULAND, which lands Blue aircraft, will need to be modified to handle tanker landings, subsequent take off scheduling, and tanker event tracts for landings.

Subroutine FUELCHK needs to be modified for interceptors. It must include logic that will help interceptors to decide where to refuel - at a tanker or at an air base.

Subroutines TNKINIT and TNKCALL will be added to the preprocessor. TNKINIT will set up the data base for anchors and TNKCALC will calculate the template flight plans for tanker flights to each anchor.

New subroutine CHZANCR will be added to help FUELCHK decide if an interceptor flight can find a anchor, and if so, which anchor to use.

New subroutine FLTANCR will process interceptor and tanker actions when a new flight arrives at an anchor hex

New subroutine TANCHOR will make up a new event module that will handle anchor processes: flight queuing, flight refueling, tanker allocations to flights, and tanker decisions about returning to the airbase.

New subroutine FLTFINR will process aircraft decisions when a flight finishes refueling at a tanker. The flight will go into orbit awaiting CRC orders, unless the flight is autonomous.

3) Data Base Modifications

Two DATFILE structure blocks will be modified. ACDB (aircraft data base) will get two new words lone also used by the BUR (enhancement) and three aliases variable names not described are explained under "User Input Modes". Also, SB will get some aliases.

ACDB

alias for ACQRANGE is TANKABTIME

alias for RADARCS is TANKFLTSPEED

alias for ATTACKRADIUS is TANKORBSPEED

add - to new words, words 12 and 13:

12A - dummy, used by BUR as PBURHEX

12B - TANKFLG (-1=tanker, 0=non-BUR, 70=BUR) also called BURFLAG by BUR enhancement

13A - TANKREF (for interceptors)

13B - REFTIME (for interceptors)

PROFILED BLOCK

alias for ALTCREN is ALT2AB

alias for ALTOTGT is ALTANCR

SB

The SB block will have the following aliases:

Word 4A - PANCHORSB - for anchor SB's, pointer to next anchor SB.

PANCHORSB - for tankers, pointer to its assigned anchor SB.

PTANCRSTAT for air bases, temporary pointer to anchor status board.

Three new MIDAS blocks will be added to constitute a new data structure for anchors. The blocks, and their relationships, are described on the next few pages. (see Figure 12)

The three new MIDAS blocks are defined below:

INKANCRSTATUS

Status board for anchors.

Word 1A - PANCRSB - pointer to SB for this anchor

B - TNEEDED - number of tankers needed in anchor hex

2A - PTNKLST - pointer to TANKERLIST, lists all tankers currently in anchor hex

b - NUMTANK - number of TANKERLIST blocks

3A - PANCHRQ - pointer to TANCHRQ, list of blocks, one block for each flight waiting to be refueled at this anchor

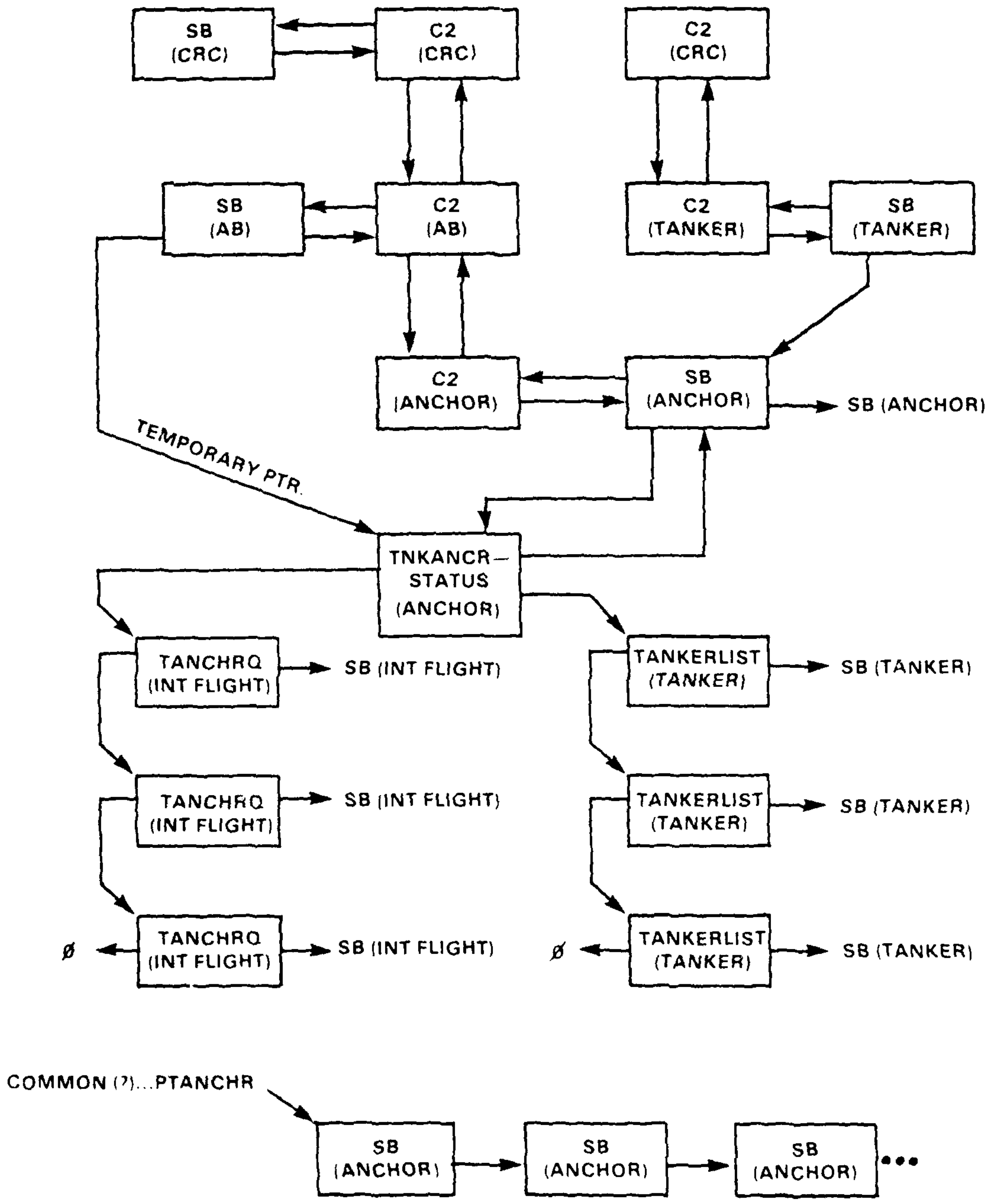
B - NUMFLTS - number of flights in queue

4A - TABTIME - tanker travel (time from AB to this anchor (seconds)

B - TTOFINT - tanker takeoff intervals for this anchor (seconds)

5A - TOTTANK - number of tankers assigned to this anchor, in various flight stages

B - NOSCRMB - number of tankers scrambled to this anchor.



4368/79W

Figure 12. Anchor Structures

TANKERLIST

List of one block for each tanker actively in the hex.

Word 1A - PNEXT - pointer to next TANKERLIST block

B - PTANKSB - pointer to tanker SB block

2A - TIMEIN - time at which the tanker entered the hex

B - TIMERLD - time tanker has spent reloading since being in the hex

3A - FUELIN - hexes of fuel in tanker when it entered the anchor hex

B - FUELRLD - hexes of fuel transferred to interceptors.

4A - PFLTSB - pointer to flight currently being refueled

B - NOUSE - not used

TANCHRQ

List of one block for each flight in the anchor queue.

Word 1A - PNEXT - pointer to next flight in queue, points to TANCHRQ

B - PFLTSB - pointer to SB for flight.

4) Common Blocks Modifications

The variable PTANCHR needs to be added to an accessible common block. PTANCHR will point to a list of SB blocks for all anchors.

5) New Events

Three new events need to be added to the simulation:

(1) Flight finished refuel from tanker (subroutine FLTFINR)

(2) TANKER Anchor Processing (subroutine TANCHR)

(3) Flight arrives at anchor hex subroutine (FLTANCR)

6) Flight Action Codes

Action code 4 will need to handle tankers orbiting in a hex. Action code 12 will be added to handle flights arriving at an anchor hex.

c. Tanker Calculations

This section describes how the raw data on tankers, anchors, and airbases is translated into tanker scheduling. The products of these calculations are 2 numbers: tanker trade off intervals and number of tankers needed to assign to an anchor. When a Blue scramble is ordered, tankers will take off at the pre-determined intervals until the required number of tankers has been launched.

GIVEN:

MAXIRT	-	longest refuel time of any interceptor seconds)
MAXFLLD	-	fuel load for that interceptor (hexes)
TANKFCR	-	tanker fuel consumption rate (hexes/second)
TABTIME	-	time it takes tanker to refuel at air base
MAXFUEL	-	fuel load (incl. cargo) of tanker
TNEEDED	-	tankers needed at each anchor hex
locations of airbases and corresponding anchor		
TANKFLTSPEED	-	speed of tanker from AB to anchor hex.

COMPUTATIONS:

MAXTRFC	=	(TANKFCR * MAXIRT) + MAXFLLD
	=	tanker refueling consumption during maximum refuel
Compute FUEL2AB		fuel consumption from AB to anchor hex (tanker)
AB2ANCR	-	distance (AB to anchor hex) computed by hex code
TIME2AB	=	AB2ANCR/TANKFLTSPEED (seconds)
FUEL2AB	=	TIME2AB * TANKFCR
MAXNORL	=	(MAXFUEL -(2* FUEL2AB))/MAXTRFC
	=	reloads possible in one flight (rounded down)
TOTRELT	=	MAXNORL * MAXFCR = total reloading time (or time in anchor hex)

TOTAL = TOTRELT + TABTIME + (2 TIME2AB)=total cycle time

TOTTANK = TNEEDED * (TOTAL/TOTRELT) (rounded up)

= total number of tankers needed for assignment to the anchor hex

TTOFINT = TOTAL/TOTTANK = take off intervals for tankers (seconds)

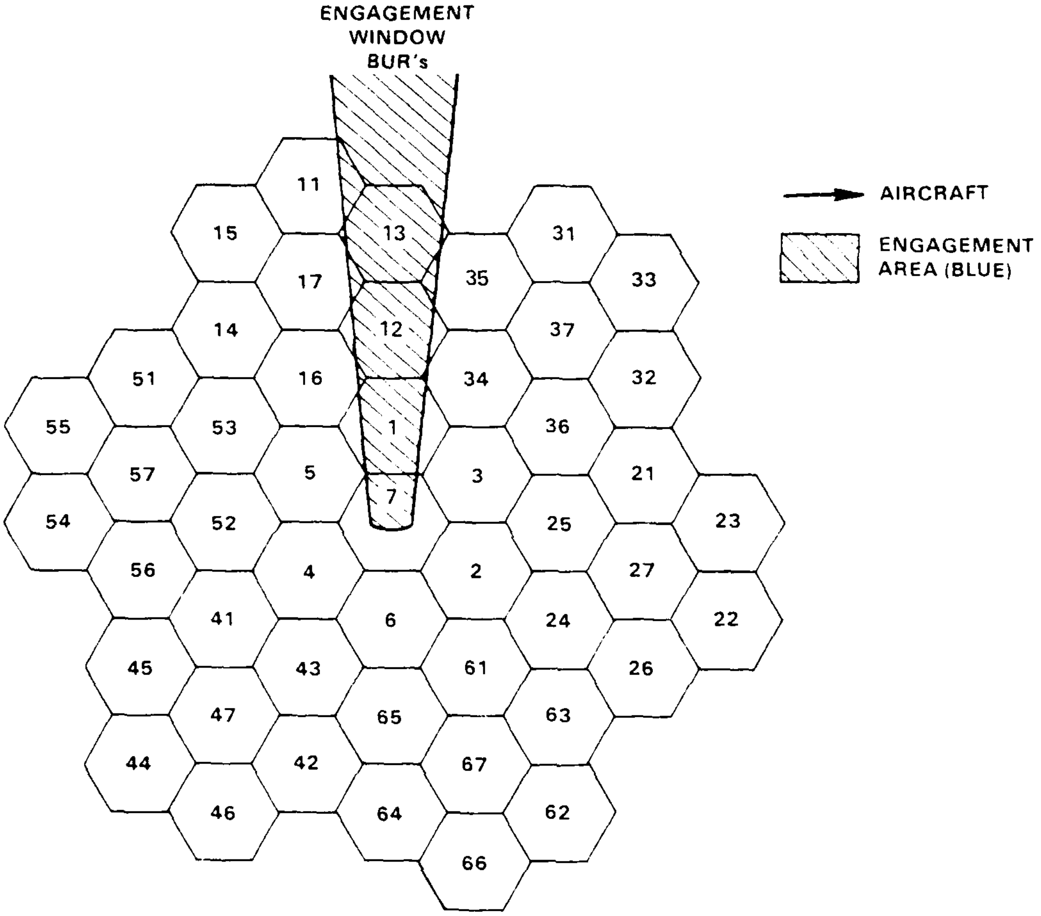
d. Beyond-Visual-Range Air-To-Air Engagement

1) Description

The purpose of this enhancement is to allow for both Red and Blue aircraft to have BVR capability for both detection and engagement. Aircraft with BVR capability will operate the same as non-BVR capable aircraft with the exception that the BVR's will have an additional window for engagement and detection. The BVR capable aircraft will maintain the same visual range as the non-BVR capable aircraft; the new BVR engagement and detection area will be an addition. However, the user will control whether or not the BVR capable aircraft have dogfight capability as well as BVR capability.

The new BVR detection and engagement area will be input by the user in the form of a template hex engagement pattern for each aircraft type. The engagement hex pattern will consist of a list of relative hex locations that are to be searched in a specified order when seeking an engagement. These hex locations are relative to the aircraft's position. The aircraft's own hex will always be searched first, followed by the hexes in the template pattern. The first enemy aircraft found in this search pattern will be immediately engaged.

The probability of kill in BVR engagements will be determined the same as before, using a random number and a Pk value that is specific to the missile type used. The probability of misidentification will also be determined as before, but BVR capable aircraft and non-BVR capable aircraft will have separate probabilities. Both of these probabilities will be input by the user.



FOR PURPOSES OF THE TEMPLATE, ALWAYS USE RELATIVE HEX LOCATIONS, ASSUMING THAT THE AIRCRAFT IS IN HEX 7, SURROUNDED BY HEXES 1-6, AND HEADING TOWARD HEX 1. HEX 7 NEED NOT BE INCLUDED IN THE HEX LIST.

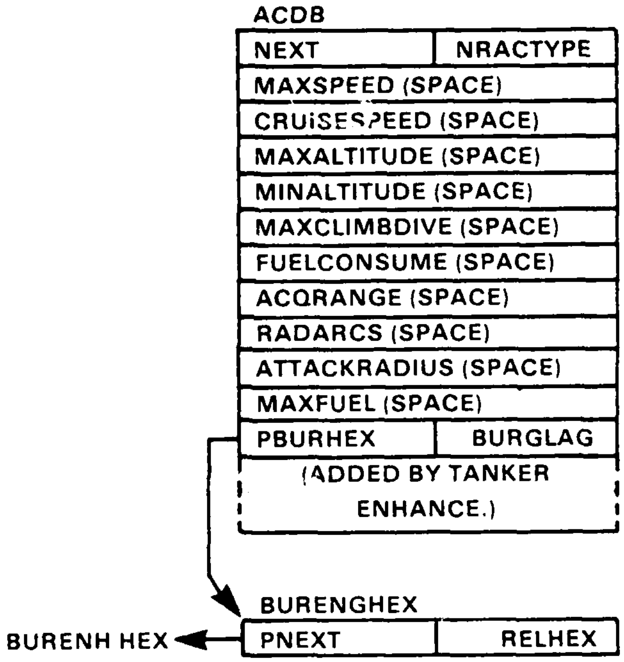
SAMPLE HEX TEMPLATE LIST: 1, 12, 13

MEANING: THE AIRCRAFT CAN ENGAGE IN HEXES 1, 12 and 13.

4368/79W

Figure 13. Sample BVR Hex Engagement Template

TABLE I. MIDAS TABLE CHANGES



4368/79W

e. Software Modifications (BVR's)

The enhancement described above will require the following software modifications:

(1) the modification of two subroutines

(2) the addition of two subroutines

(3) the modification of one MIDAS data structure

(4) the addition of one MIDAS data structure

(5) the addition of one common block

1) Subroutine Changes

The data input subroutine OTHRDAT will be modified to allow for three new types of input:

(1) the definition of BVR capable aircraft

(2) the definition of BVR capable aircraft

(3) the two (BVR and non-BVRV misidentification probabilities

This will require four new input variables, described below:

BURFLAG - a flag that indicates whether or not an aircraft type has BVR capability.

BURHEXS - a list of relative hexes that make up the template for the BVR engagement window, input by aircraft type

RMISID1 - misidentification probability for non-BVR engagements. stored in new common block PMISID

RMISID2 - misidentification probability for BVR engagements. stored in new common block PMISID

In the Ponder module, subroutine TFLYCRC (fly and CRC ponders) will have to be modified to include new logic for BVR capable aircraft. Basically, when a BVR capable aircraft does not find an engangement possibility in its own hex, the new logic will include a call to BVRTHNK, the new subroutine that searches the BVR engagement window.

New Ponder subroutine BVRTHNK will be added to handle BVR detection and engagement. This routine will be similar to AIRTHNK, which handles the same for non-BVR aircraft. When BVRTHNK decides an aircraft should fire a BVR missile, it will schedule a new event - BVR missile arrives at target - which will be handled by BVRMAT.

New Ponder subroutine BVRMAT will be the BVR version of dogfite, with the obvious differences. BVRMAT will decide the outcome of a BVR missile firing, and act accordingly.

2) Data Structure MODS

The MIDAS data block ACDB (aircraft date base - one block for each aircraft type) will have 2 new fields added to it:

PBVRHEX - pointer to new block BURENGHEX, which will list the hex template for the BUR engagement window

BURFLAG - a flag indicating whether or not an aircraft type has BUR capability.

NOTE: I suggest that these two fields occupy one word such that ADDBLOK can use the word as a buffer when adding BVRENGHEX blocks. In this way, BVRFLAG = 0 would mean no BVR capability. The BVRFLAG field can also be used to identify tankers by using a negative number.

The new MIDAS data block BURENGHEX will contain one relative HEX address, and a pointer to the next BURENGHEX block. A list of these one word blocks will constitute the hex engagement window template for an aircraft type.

3) Common Block MODS

The new common block PMISID will contain the two probabilities ror misidentification: RMISID1 (non-BUR) and RMISID2 (BUR). These must be real numbers.

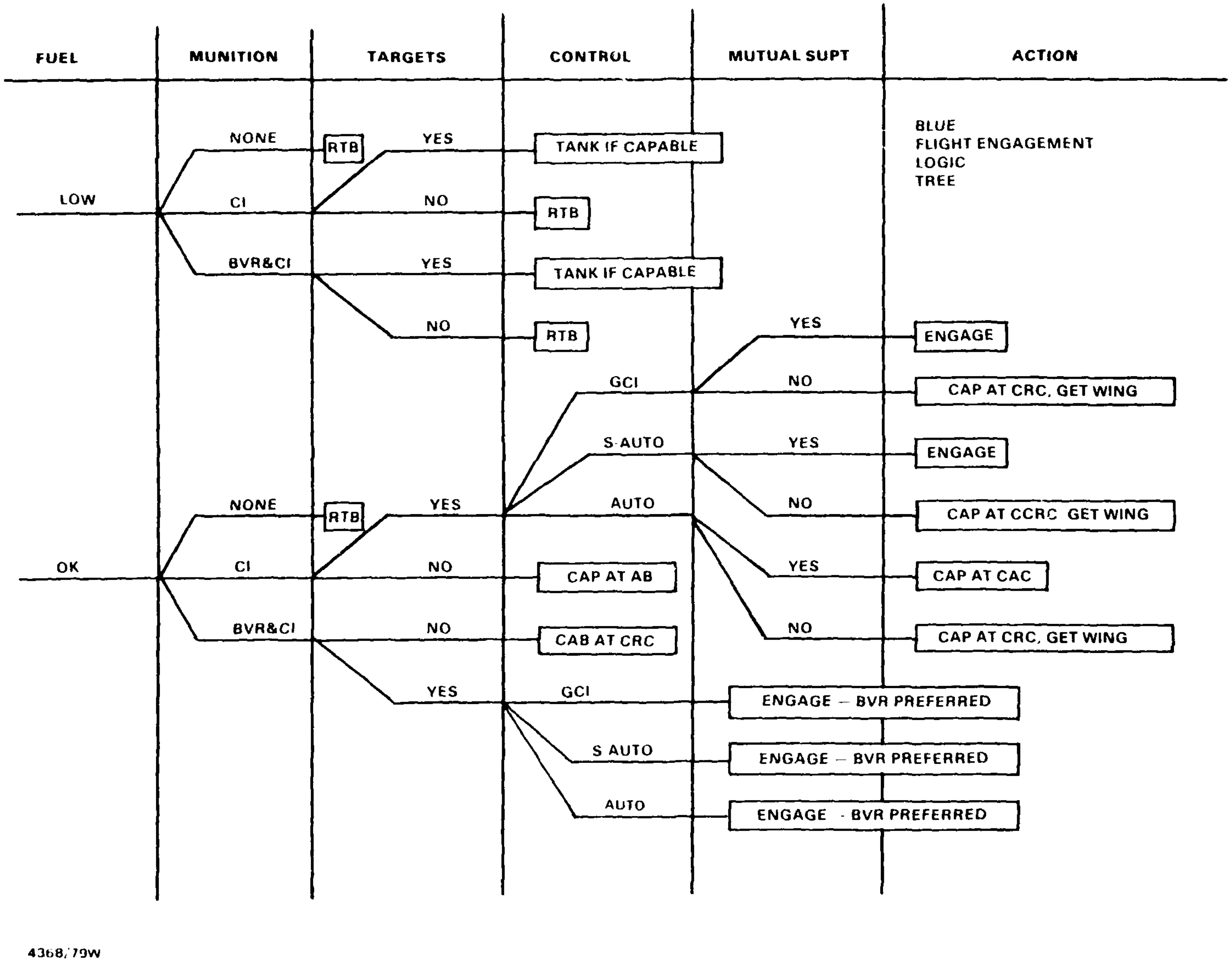
2. Disengagement Logic

a. Description

This enhancement attempts to adjust the model to follow the disengagement logic shown on the next page. The model currently follows this logic with the exception of the mutual support logic.

To include mutual support logic, the dogfight engagement logic must be modified. It will be modified so that Blue flights will not engage alone, but will have at least one other flight in mutual support. If mutual support cannot be found when a target is detected, the Blue interceptor will set up a CAP at the CRC's order.

TABLE 2. BLUE FLIGHT ENGAGEMENT LOGIC TREE



b. Software Modifications

The enhancement described above will only require the modification of one subroutine.

Subroutine DOGTHNK, the DOGFITE PONDER routine, will have an in-line IF test to determine if mutual support is available or not. If not, a CAP is set up at CRC order.

D. MOBILE SAM UNITS

This enhancement is designed to allow simulation of two types of hypothetical SAM units which may be developed in the near future. The first type is highly mobile (assumed to be mounted on some type of rough terrain vehicle) and may be reassigned from one BOC area to another at intervals set by the user. The second type is somewhat less mobile (assumed to require breakdown and reassembly) and may only be moved from one allocation area to another between raids. The designs for these two mobile SAM unit types also represent different approaches to the problem of resource reallocation in an air defense environment. Both designs are discussed in the following sections.

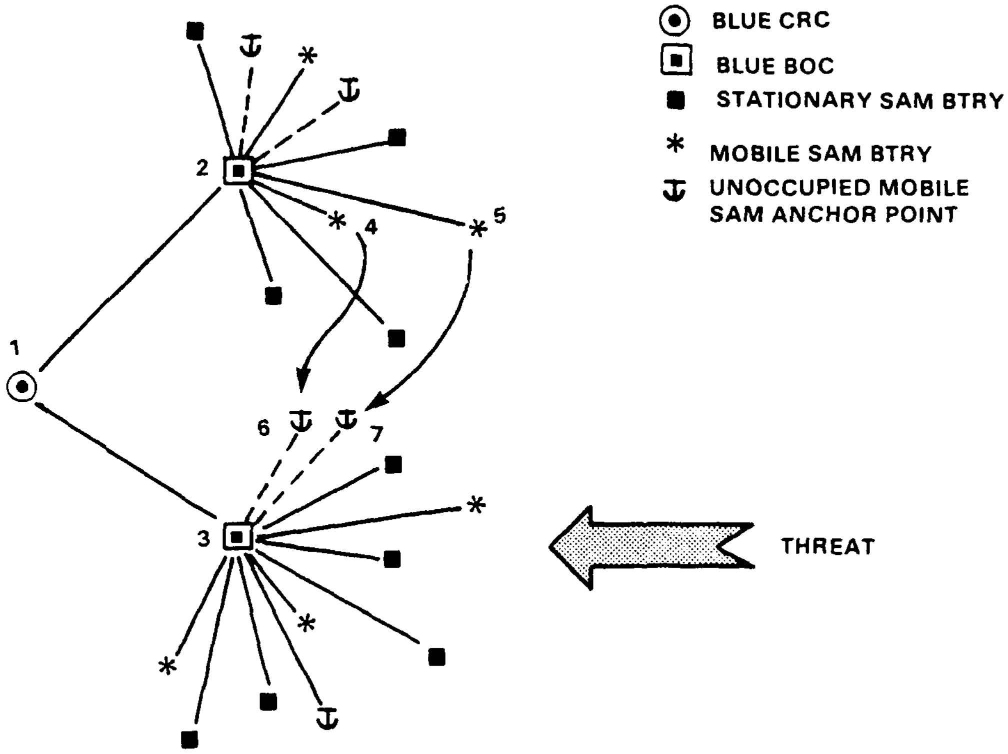
1. High Mobility SAM Units

a. Description

The enhancement is designed to allow simulation of hypothetical mobile SAM batteries which may be developed in the near future. The basic mechanics of this enhancement are very similar to the overall threat perception and resource allocation algorithm used for interceptor reassignment among CRC's by the ATAF. The primary difference is that mobile SAM's will be reallocated among BOC's by their commanding CRC.

The user will specify the operational characteristics of each mobile SAM unit type in the data base file. A series of mobile SAM anchor points and actual mobile SAM locations will then be specified using new user oriented input language (UOIL) sentences. Reallocation constraints for each BOC will also be input by the user. The constraints will include the maximum number of mobile SAM's to be reallocated in a given reallocation event, maximum SAM units per anchor point, and travel speed of the SAM units.

The CRC will respond to overloading of a given BOC by attempting to reallocate mobile SAM's from BOC's where a surplus of mobile SAM's exists. The reallocation algorithm will use the forecasted demand technique discussed in Section B to anticipate threats against BOC's. It will also attempt to minimize SAM travel distance. Figure 14 illustrates the mobile SAM concept.



MOBILE SAM's 4 AND 5 REALLOCATED TO BOC 3 AT ANCHOR POINTS 6 AND 7

4368/79W

Figure 14. Mobile SAM Reallocation Example

b. Software Modifications

This enhancement will use modified versions of the software and logic outlined in Section B and Appendix B. SAM battery movement mechanics will be handled through modifications to the flight movement mechanics routines. With the exception of their movement capabilities, mobile SAM's will be treated in exactly the same manner as stationary SAM's. Because of its similarity to the interceptor reallocation algorithm, no special mobile SAM PDL has been written at this time.

2. Low Mobility SAM Units

a. Description

This enhancement is designed to allow simulation of hypothetical mobile SAM batteries which may be developed in the near future. The reallocation event can only take place between raids. The reallocation algorithm will try to reallocate remaining mobile SAM's in the same relative density as before the initial raid.

The user will specify various characteristics and constraints of mobile SAM reallocation. In the DATFILE, where SAM characteristics are defined, the user will indicate whether or not a SAM type is mobile. Only the mobile SAM types can be reallocated. In the user input language file (UOIL), the user will divide all reallocatable SAMs into geographically divided groups. Also input by the user will be the travel speed of the SAM unit types that are mobile.

At the end of each raid, each geographical group of mobile SAM's will be assessed for damage. An overall survivability rate will be determined based on a combination of all groups and their damage level. This overall average of survival will be compared to the actual survival rate for each group. Groups doing better than the average will be considered supply groups, and groups that did worse than the average will be considered demand groups. The amount of mobile SAM's in surplus in each supply group will be determined by the number that could be taken away and still have at least the average survival rate. For demand groups, the demand will be the number of units needed to reach the average survival rate.

Once it is determined how many mobile SAM's each group has in surplus or demand, then the reallocation can occur. The group with the highest demand will be given mobile SAM's from the nearest supply group. Then the next highest demand group will be taken are of, and so on. In this way, the resulting scenario should have the mobile SAM's in approximately the same relative density that they were originally allocated. (See Figure 15.)

To determine which SAM's get reallocated or replace within a particular group, a priority system could be set up. The user could enter priorities for each mobile SAM within each group. SAM's with a high priority would be replaced before a SAM with a low priority. Likewise, SAM's with a high priority. This added feature could prevent important surviving SAM's from being reallocated, and thus further enhance the reality of this design.

It should also be pointed out that not all mobile SAM's need be actually mobile. The user may eliminate the possibility of certain "mobile" units being reallocated simply by leaving these units out of any of the geographical reallocation groups.

b. Software Modifications

The following software modifications will be necessary to perform the above enhancement:

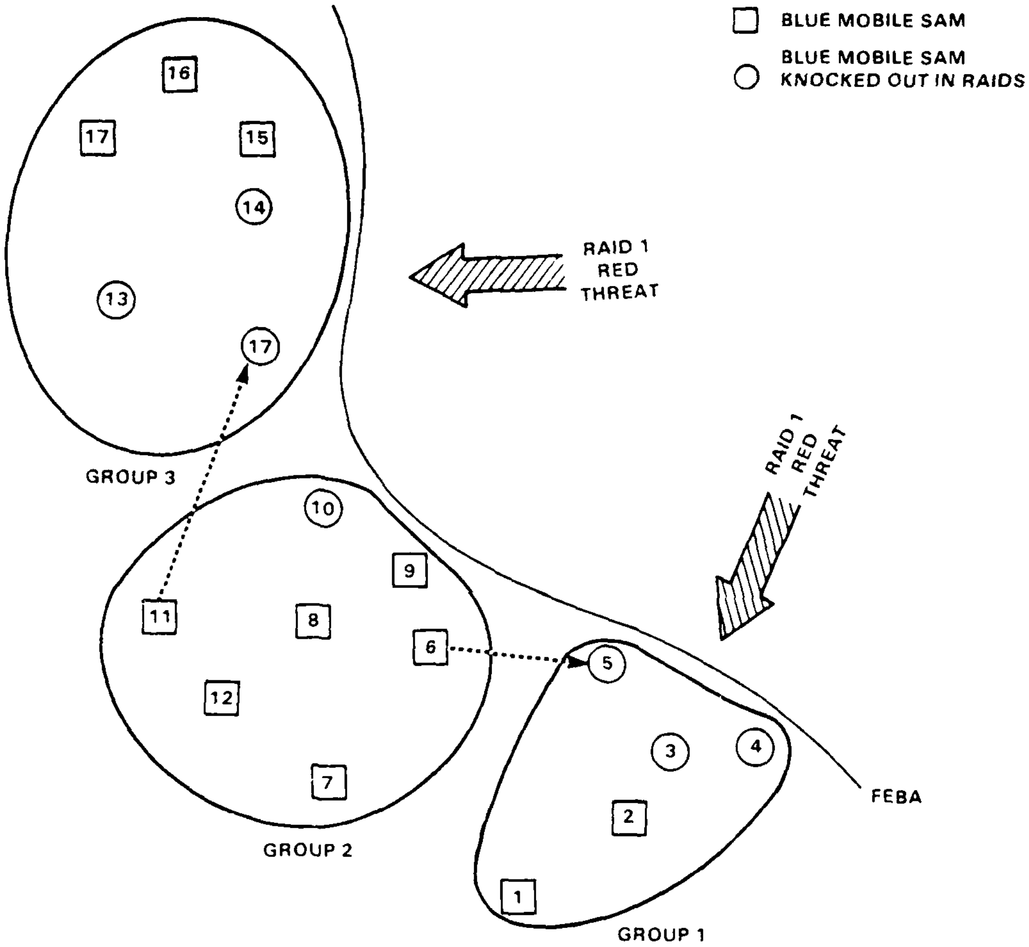
(1) modification of three existing subroutines: OTHRDAT, SEMANT, and PLAN

(2) addition of two new subroutines: RMOBSAM - reallocates mobile SAMs between raids, AMOBSAM - arrival of mobile SAM at new site

(3) the addition of two new MIDAS blocks in a new MIDAS structure: MOBLGRD - mobile SAM group header block, MOBILE - mobile SAM block

(4) the addition of one common pointer: PMOBILE - pointer to groups of mobile SAM's.

Data input subroutine OTHRDAT will need to be modified to accept changes in the air defense site data base of the DATFILE. Two fields will be added here, one to indicate whether or not a SAM is mobile,



GROUP	INITIAL	LEFT	SURVIVAL RATIO	D⁻MAND SURPLUS	ROUNDED TO
1 2 3	5 7 6	2 6 3	2 5 6 7 3 6	-1.05 +1.72 0 67	1 +2 -1
TOTAL	18	11	--	0	0

OSR - OVERALL SURVIVAL RATIO 11 18 0 61
DEMAND SURPLUS - LEFT - (INITIAL & OSR)

4368,79W

Figure 15. Mobile SAM Reallocation Post Raid 1

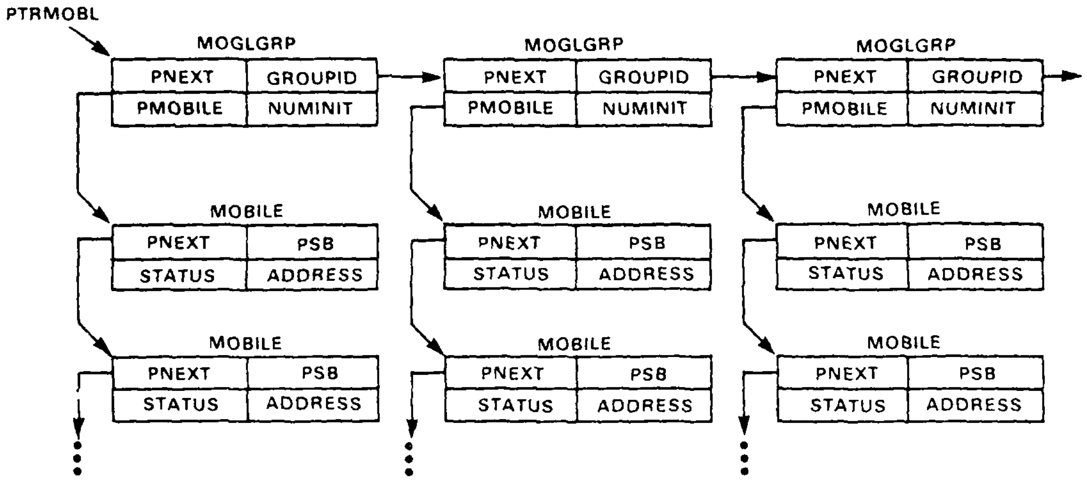
and one to indicate the speed of travel for each mobile SAM type. (see Figure 16.)

UOIL input subroutine SEMANT will be modified to accept sentences describing the mobile SAM groups, i.e. [4 BATTERY BELONGS TO 8 MOBILE GROUP]. SEMANT will also set up the group data structure.

Subroutine PLAN will be modified to initiate SAM reallocation before planning the next raid.

The two new subroutines are outlined with PDL in Appendix D.

The MIDAS additions are explained on the next page.



MOBLGRP – REPRESENTS ONE GEOGRAPHICAL MOBILE GROUP
PNEXT – POINTER TO NEXT MOBLGRP BLOCK
GROUPID – ID OF THIS GROUP
PMOBILE – POINTED TO LIST OF MOBILE SAM LOCATIONS
NUMINIT – NUMBER OF MOBILE SAM's INITIALLY IN GROUP

MOBILE – REPRESENTS EACH ORIGINAL MOBILE SAM LOCATION
PNEXT – POINTER TO NEXT MOBILE BLOCK FOR THIS GROUP
PSB – POINTER TO SB BLOCK OF SAM
STATUS – STATUS OF THE SAM, I.E., ALIVE OR DEAD
ADDRESS – ADDRESS OF THIS SAM

4368/79W

Figure 16. Mobile SAM Data Structure

E. VARIABLE SHORAD DENSITY

1. Description

The purpose of this enhancement is to allow simulation of SHORAD point defenses. The user will input a background SHORAD density and associated P(k) value for all hexes in the battle area. This background P(k) will then be applied to all flights as they traverse the hex grid. In addition, the user may specify increased SHORAD densities for as many hexes as are required to simulate point defenses. The increased SHORAD densities will then be translated to higher P(k) values and applied to flights passing through the area.

2. Software Modifications

This enhancement will require modifications to the OTHRDAT data input routine to allow SHORAD density inputs for multiple hexes and to SHRKILL SHORAD engagement routine. In addition a new word containing the SHORAD density level will be added to the Hex block data structure and the Hex creation routine GETHEX modified accordingly.

F. SURFACE-TO-SURFACE MISSILES

1. Description

This enhancement is designed to simulate the use of surface to surface missiles against NATO targets. Missiles will be allocated using the same basic attack planning specifications now used for Red penetrator aircraft. The user will specify the range of the missile types to be employed and their destructive potential. Missiles will then be allocated against specific target types in the same raid/ wave planning cycle as aircraft. The only functional difference between missiles and aircraft is that missile will fly on a ballistic trajectory that will carry them along a straight ground path to their target. Attack corridors will be ignored, SHORAD attritions will not be applied. However, missiles may be engaged by Blue SAM defensive units. Red missiles will be launched from designated launch bases which will function in much the same way as air bases now operate.

2. Software Modifications

This enhancement will require very few modifications to the existing software. In essence, surface to surface missiles and their launch bases can be mechanically treated as just another type of flight and air base. The ACFRAG flight path generation rougine in the PLAN module will be modified to generate orders for a ballistic flight path with no return. The OTHRDAT data input routine will have to be altered to "tag" missiles as a special type of flight.

APPENDIX A

*SEGMENT (ASSIGNMENT MODULE)
SUBROUTINE ASSIGN

This routine simulates communications among CRC's by controlling the degree of multiple assignments on a target. Probability levels can be placed on the true recognition of the assignment, and the degree of misidentification by IFF. In addition, engagement controls by the CRC can be levied here. Currently all perceived unassigned enemy aircraft are candidates for assignment. Also, a priority on whether interceptors or battalions get the first attempt against a weighting factor. If interceptors cannot be assigned to a target, an attempt will be made to assign the target to a battalion. If a battalion cannot be assigned to a target, interceptors will not be given a chance to make an assignment to avoid a logical loop.

commons, etc ____________

DECLARE PTGTSB=SB, ICRC+SB

```
          IOBON=Ø
          PTGTSB=MSG
          ICRC=NEHMEN
C
C         *IF (TARGET ALREADY ASSIGNED) THEN
           IF ($PTGTSB.PARCFTSTAT.ASGNFLG$.EQ.Ø) GO TO 3ØØ
C
C              *IF (ASSIGNMENT TRUTH NOT RECOGNIZED) THEN
                IF ( RANF(RSEED).GT. ARL1) GO TO LØØ
C
C                   *DO NOT ATTEMPT ASSIGNMENT
                     IFLG=Ø
                     GO TO 2ØØ
C
C              *ELSE
1ØØ             CONTINUE
C                   *ATTEMPT ASSIGNMENT
                     IFLG=1
C
C              *END IF
2ØØ             CONTINUE
                TO TO 5ØØ
C
C         *ELSE (TARGET IS NOT ASSIGNED)
3ØØ        CONTINUE
C
C              *IF (NONASSIGNMENT TRUTH IS RECOGNIZED) THEN
                IF ( RANF(RSEED).GT.ARL2) GO TO 4ØØ
C
C                   *ATTEMPT ASSIGNMENT
                     IFLG=1
                     GO TO 5ØØ
```

```
C
C              *ELSE
4ØØ             CONTINUE
C
C                   *DO NOT ATTEMPT ASSIGNMENT
                     IFLG=Ø
C
C                 *END IF
C              *END IF
5ØØ             CONTINUE
C              *IF (ASSIGNMENT IS TO BE ATTEMPTED) THEN
                IF (IFLG.EQ.Ø) TO TO 8ØØ
C
C                 *IF (TARGET IS BLUE) THEN
                   IFF ($PTGTSB.PC2.SIDE$.GT.1) TO TO 6ØØ
C
C                        *APPLY IFF CRITERIA TO FRIENDLY AIRCRAFT
                          IF=Ø
                          IF (RANF(RSEED).GT.FFLVL1) IFF=1
                          TO TO 7ØØ
C
C              *ELSE
6ØØ             CONTINUE
C
C                        *APPLY IFF CRITERIA TO ENEMY AIRCRAFT
                          IFF=1
                          IF (RANF(RSEED).GT.FFLVL2) IFF=Ø
C
C              *END IF
7ØØ             CONTINUE
                GO TO 9ØØ
C
```

```
C              *ELSE (ASSIGNMENT NOT TO BE ATTEMPTED)
8ØØ             CONTINUE
C
C                   *SET NO ASSIGNMENT FLAG
                     IFF = Ø
C
C              *END IF
9ØØ             CONTINUE
C
C              *IF (TARGET IS THOUGHT TO BE ENEMY) THEN
                IF (IFF. EQ. Ø) GO TO 12ØØ
C
C                   *INCLUDE (ENGAGEMENT CONTROL
C                    IMPOSED BY CRC)
                     IENG = 1
C
C                   *IF (TARGET IS TO BE ENGAGED) THEN
                     IF (IENG.EQ.Ø) GO TO 12ØØ
C
C                        *DETERMINE WHICH SYSTEM GETS
C                         ASSIGNMENT ATTEMPT FIRST
                          IFLG = Ø
                          IF (RANF(RSEED).GT.APRTY) IFLG = 1
C
C                        *IF(INTERCEPTORS GET FIRST
                          ASSIGNMENT ATTEMPT) THEN
                          IF (IFLG.EQ.Ø). GO TO 1ØØØ
C
C                             *INCLUDE (INTERCEPTOR
C                              ASSIGNMENT ATTEMPT)
                               CALL INTASIN
                               GO TO 11ØØ
```

```
C              *ELSE (BATTALIONS GET ASSIGNMENT ATTEMPT)
1ØØØ            CONTINUE
C
C                   *LOCATE TARGET PERCEPTION BLOCK
                     CALL FINDBLK ($ICRC.PSDB.PSEE$,
     t               PTGTSB,2,ICQUARY)
C
C                   *INCLUDE (BATTALION ASSIGNMENT
                     ATTEMPT)
                     CALL BTNASIN (ICQUARY)
C
C              *END IF
11ØØ
C                 *END IF
C              *END IF
12ØØ            CONTINUE
C
               *INCLUDE (RECORD HOW CRC MADE ASSIGNMENT)
                CALL CADR (ICRC)
C
C         *END SEGMENT
9ØØØ       CONTINUE
           RETURN
           END
```

APPENDIX B

*SEGMENT (ATAF INTERCEPTOR REASSIGNMENT)
SUBROUTINE ATAF

This routine reallocates interceptors among CRC's based on the preceived ability of the CRC's to assign incoming Red flights. It is called as a result of an reallocate event. Reallocate events occur at fixed intervals set by the user or when a CRC exceeds its user defined overload threshold.

CRC's that could not assign incoming Red flights during the preceding reallocation interval are considered to be demand CRC's. Demand is measured in terms of the number of flights required to cover the unassigned Red flights.

CRC's that had aircraft left over after assigning all incoming Red flights are considered to be supply CRC's. Aircraft on air bases commanded by supply CRC's may be reallocated to air bases commanded by demand CRC's.

Aircraft are reallocated to demand CRC air bases on the basis of the CRC's demand level, the ability of its' air bases to service various aircraft types, the availability of various aircraft types at supply CRC's, and the distance of supply air bases from demand air bases.

This routine assumes the existence of a CRC assignment distribution list created and maintained by the new subroutines CADR and UCSCL. After traversing the above list this routine creates two new temporary lists - one for supply CRC's and one for demand CRC's.

```
*COMMONS
*MIDAS DECLARATIONS

*DO UNTILE (END OF CRC ASSIGNMENT DISTRIBUTION LIST)

     *IF (THIS IS A DEMAND CRC) THEN
          *INCLUDE (CREATE AND POSITION NEW BLOCK IN
                    THE SORTED DEMAND CRC LIST)
                    HIGHRST DEMAND CRC AT TOP
     *ELSE
          *INCLUDE (CREATE AND POSITION NEW BLOCK IN THE
                    SORTED SUPPLY CRC LIST)
                    HIGHEST SUPPLY OR PRIORITY CRC AT TOP
*ENDDO
*DO UNTILE (END OF DEMAND CRC LIST OR SUPPLIES EXHAUSTED)
          *INCLUDE (SORT DEMAND CRC AIR BASE LIST IN DECREASING
                    ORDER OF DAMAGE LEVEL)
          *DO UNTILE (END OF DEMAND CRC AIR BASE LIST)
             *DO UNTILE (END OF AIRCRAFT TYPES ON AIR BASE LIST)
                         MAXIMUM SERVICE CAPACITY REACHED)
               *DO UNTILE (END OF SUPPLY CRC LIST OR DEMAND
                           SATISFIED)
                    *INCLUDE (SORT SUPPLY AIR BASES IN DECREASING
                              ORDER OF DISTANCE FROM THE DEMAND
                              AIR BASE UNDER CONSIDERATION)
                    *DO UNTILE (END OF SUPPLY AIR BASE LIST OR
                     DEMAND SATISFIED)
                         *DO UNTILE (END OF AIRCRAFT TYPES ON
                                     AIR BASE LIST OR DEMAND
                                     SATISFIED)
                              *IF (SUPPLY AC TYPE. EQ. DEMAND AC
                                   TYPE AND MINNO, MAXNO, MAXRAL
                                   CONSTRAINTS MET) THEN
```

```
                               *LAUNCH MAXIMUM NUMBER OF
                                FLIGHTS THAT CAN BE BUILT
                                FROM AVAILABLE AIRCRAFT
                               *ATTACH NEW FLIGHT'S AIR BASE
                                POINTER TO DEMAND AIR BASES
                               *DECREMENT DEMAND LEVEL
                                FOR CURRENT CRC
                            *ENDIF
                        *ENDDO
                    *ENDDO
                *ENDDO
            *ENDDO
        *ENDDO
    *ENDDO
    *INDLUDE (RELEASE TEMPORARY STRUCTURES)
*END SEGMENT
```

```
*SEGMENT (SORT SUPPLY AIRBASES IN DECREASING ORDER OF
          DISTANCE FROM THE DEMAND AIR BASE)
SUBROUTINE DSTSRT (PDABSB)
```

This routine sorts the supply CRC's subordinate airbase CRCSUBROD block list in decreasing order of distance from the demand air base currently under consideration for reallocation.

The distance between air bases is calculated using the routine HEXDIST. The resulting distance measure is stored in a new field in the CRCSUBORD block.

```
     PDABSB =  POINTER TO DEMAND AIR BASE SR
               BLOCK.  USED TO FIND HEX ADDRESS
               OF THE DEMAND AIRBASE

*COMMONS
*MIDAS DECLARATIONS

*GET ADDRESS OF DEMAND AIR BASE
     DHEX = $PDABSB.ADDRESS$

*DO UNTILE (END OF SUPPLY AIR BASE LIST)

     *GET ADDRESS OF SUPPLY AIR BASE
      sHEX = $PABLST.ADDRESS$

     *INDLUDE (DISTANCE FROM DEMAND TO SUPPLY AIR BASE)
      CALL HEXDIS (DHEX, SHEX, DIST)

     *INSERT DISTANCE INTO CRCSUBORD BLOCK OF SUPPLY AIR BASE
      $PABLST.DISTANCE$ = DIST
*ENDDO
*PERFORM BUBBLE SORT OF CRCSUBORD BLOCK
*END SEGMENT
```

```
*SEGMENT (CREATE SORTED DEMAND OR SUPPLY CRC LIST)
 SUBROUTINE SOSORT (MODE)
```

This routine adds an <u>ALLOCATE</u> block to either the supply or demand CRC list. The list is then sorted. The supply list is sorted by demand level while the demand list is sorted by supply level or user input priorities.

```
     MODE + DEMAND/SUPPLY LIST FLAG
            1 = DEMAND LIST SORTED ON DEAMND LEVEL KEY
            2 = SUPPLY LIST SORTED ON SUPPLY LEVEL KEY
            3 = SUPPLY LIST SORTED ON PRIORITY LEVEL KEY
```

The insertion sort method is used to insert an new allocate block with the specified key into the supply or demand list in such a way that the resulting list is also ordered on the specified key.

```
*COMMONS
*MIDAS DECLARATIONS

     *CREATE NEW ALLOCATE BLOCK AND SET VALUES
      EQUAL TO CORRESPONDING VALUES IN ITS" ASSIGNDATA BLOCK

*CASE (MODE)

     *MODE = 1
          *KEY = $ PDEMAND. DEMAND $

     *MODE = 2

          *KEY = $ PSUPPLY.SUPPLY $

     *MODE = 3
```

```
          *KEY = $ PSUPPLY. PRIORITY $
     *END CASE
     *PERFORM INSERTION SORT
*END SEGMENT

*SEGMENT (SORT DEMAND CRC AIRBASE LIST IN DECREASING ORDER OF DAMAGE
          LEVEL)
 SUBROUTINE DAMSRT
```

This routine sorts the demand CRC's subordinate airbase CRCSUBORD block list in decreasing order of damage level. A bubble sort algorithm is used to order the list so that the least damaged air bases rise to the top of the list.

Demand airbases are assigned new interceptors in order of their occurrence on this list. Therefore, the least damaged demand air bases will be given priority for reassignment of aircraft

```
*COMMONS
*MIDAS DECLARATIONS
     *PERFORM BUBBLE SORT OF CRCSUBORD BLOCKS
*END SEGMENT
```

*SEGMENT (CRC ASSIGNMENT DISTRIBUTION RECORDER)
SUBROUTINE CADR (ICRC)

This routine records how the CRC is assigning assets, whether to interceptors, battalions or being unable to assign to either.

This record is kept in a linked list of 5 word blocks pointed to by the variable PIOBON.

The purpose of this list is to allow a higher authority above CRC to a basis for decision making when reassigning assets among CRCs. An event can periodically be scheduled to do the evaluation and make the appropriate redistribution of assets. The counters can then be reinitialized as desired.

If a CRC is destroyed, the SB pointer of the CRC is set negative by destroy.

Do not reference the positive value of the CRC SB pointer since the scoreboard data structure of the CRC will have been released. This CRC no longer exists. However, the location of the dead CRC's air base list is retained for possible assignment of assets to that location.

Commons

```
      IFND = Ø
      PTR = PIOBON

C
C     *DO UNTIL (APPROPRIATE CRC BLOCK FOUND OR AT END OF LIST)
1ØØ    CONTINUE
       IF (PTR.EQ.Ø) GO TO 4ØØ
C
C          *IF (THIS BLOCK IS CORRECT BLOCK) THEN
```

```
           IF (ICRC.NE.ISPACE(PTR+1)) GO TO 2ØØ
C
C              *SET CRC BLOCK FOUND FLAG
                IFND = 1
                PCRC = PTR
                PTR = Ø
                GO TO 3ØØ
C
C         *ELSE
2ØØ        CONTINUE
C
C              *GET NEXT BLOCK IN LIST
                PTR = ISPACE(PTR)
C
C         *END IF
3ØØ        CONTINUE
C
       *END DO
        GO TO 1ØØ
4ØØ     CONTINUE
C
C    *IF (NO BLOCK WAS FOUND) THEN
      IF (INFND.NE.Ø) TO TO 5ØØ
C
C              *CREATE A BLOCK FOR THIS CRC AND LINK IT
                CALL GIMME (PTR,5)
                ISPACE (PTR) = PIOBON
                ISPACE (PTR=1) = ICRC
                PIOBON = PTR
                PCRC = PTR
C
C         *END IF
```

```
5ØØ        CONTINUE
C
C         *RECORD CRC ASSIGNMENT RESULT BY UPDATING COUNTER
           ISPACE (PRCR + 2 +IOBON) = ISPACE (PCRC + 2 + IOBON) + 1
C
C    *END SEGMENT
6ØØ   CONTINUE
      RETURN
      END

*SEGMENT (UPDATE CRC STATUS ON CADR LIST)
 SUBROUTINE UCSCL (PCRC,PHEX)
```

A CRC has been destroyed. This routine notes this fact by modifying the appropriate block for this CRC in the CRC assignment distribution list. The SB pointer of the CRC is set negative in the second word of the data, structure. The CRC's air base list, pointer is set.

Commons

```
     *DO UNTIL (APPROPRIATE BLOCK FOUND OR LIST EMPTY)
      PTR = PIOBON
1ØØ   CONTINUE
      IF (PTR.EQ.Ø) GO TO 4ØØ
C
C         *IF (THIS BLOCK IS CORRECT BLOCK) THEN
           IF (PCRC.NE.ISPACE(PTR = 1)) GO TO 2ØØ
C
C              *UPDATE BLOCK CONTENTS
                ISPACE (PTR + 1) = -PCRC
                ISPACE (PTR + 2) = -PHEX
                PTR = Ø
                GO TO 3ØØ
C
C         *ELSE
2ØØ        CONTINUE
C
C              *GET NEXT BLOCK IN LIST
                PTR = ISPACE (PTR)
C
C         *END IF
3ØØ        CONTINUE
C
C         *END DO
           GO TO 1ØØ
4ØØ        CONTINUE
C
C    *END SEGMENT
5ØØ   CONTINUE
      RETURN
      END
```

APPENDIX C

This appendix includes PDL for the interceptor operations enhancements.

1. Tankers
2. BVR Target Acquisition
3. Disengagement Logic

1. Tankers PDL

```
*SEGMENT (SEMANT-SEMANTIC PROCESSING)
      ●
      ●
      ●
****________________________________***
*number ANCHOR DESIRES number Tankers*
***_________________________________***
*GET numbers from sentence
*INCLUDE (TNKINIT - Tanker initial calculations).
      ●
      ●
      ●
*END SEGMENT (LEMANT)
```

```
*SEGMENT (TNKINIT-INITIAL TANKER PROCESSING)
     *FIND ANCHOR IN C2 TREE
     *IF (NOT FOUND) THEN
          *INDICATE ERRONEOUS INPUT
          *EXIT SEGMENT (TNKINIT)
     *END IF
     *ALLOCATE TNKANCR STATUS BLOCK
     *SET AB SB TO POINT TO TNKANCR STATUS
     *PUT ANCHOR'S SB INTO ANCHOR LIST (TOP OF LIST)
     *INCLUDE (TNKCALC - PERFORM ANCHOR CALCULATIONS)
     *FILL TNKANCRSTATUS BLOCK WITH DATA
*END SEGMENT (TNKINIT)
```

```
NOTE:  See section on tanker calculations
     *SEGMENT (TNKCALC - TANKER - ANCHOR CALCULATIONS)
          *GET ACCESS TO:
               MAXIRT        - Compare REFTIME in ACDB blocks
               MAXFLLD       - ACBD block that MAXIRT came from
               TANKFCR       - ACBD for tanker
               TABTIME       - ACDB for tanker
               MAXFUEL       - ACDB for tanker
               TNEEDED       - Parameter
               TANKFLTSPEED - ACDB for tanker
          *COMPUTE TOTTANK - TOTAL TANKERS NEEDED
          *COMPUTE TTOFINT - INTERVAL TIME FOR TAKEOFFS
     *END SEGMENT (TNKCALC)
```

```
(FROM BEN'S ROUTINE)
     *SEGMENT (BLUTKOF - BLUE TAKE OFF)
          *IF (TANKERS) THEN
               *GET PTR TO TNKANCRSTATUS BLOCK FROM AB SB.
          *ENDIF
          *LOOP = Ø
          *DO UNTIL (LOOP = Ø) - Do loop once for non-tankers, for non-
                                  tankers, until all processed
               **IF (ANY A/C AVAILABLE FOR LAUNCH), THEN
                    ●
                    ●
                    ●
                    *INCLUDE (CRFLTML - CREATE FLIGHT MINUS UOL, PL)
                    *WRITE LAUNCH MESSAGE, INDICATE TANKER OR INTERCEPTOR
                    ●
                    ●
                    ●
                    *SCHEDULE AIRCRAFT LAND
                    *IF (TANKER) THEN
                         *SCHEDULE FLIGHT BACK TO AB
                         *SCHEDULE END OF ANCHOR ORBIT
                         *SCHEDULE ANCHOR ORBIT
                    *END IF
                    *IF (CRC IS ALIVE) THEN
                         *IF (TANKER) THEN
                              *SCHEDULE 1ST LEG TO ANCHOR HEX
                         *ELSE
                              *SCHEDULE 1ST LEG TO CRC ADDRESS
                         *END IF
                         *NOTIFY CRC OF TANKOFF
               *ELSE (AUTONOMOUS OPERATIONS)
                    *IF (TANKER) THEN
                         *SCHEDULE 1ST LEG TO ANCHOR HEX
                    *ELSE
                         *SCHEDULE ORBIT AROUND AIR BASE
                    *END IF
               *END IF
               o
               o
               o
               *SCHEDULE 1ST FLY MANEUVER IN 5 SECONDS
               *IF (AIRCRAFT ON HAND) THEN
                    *IF (TANKER) THEN
                         *IF (MORE TANKERS NEEDED AT CURRENTLY PROC-
                              ESSED ANCHOR) THEN
                              *SCHEDULE APPROPRIATE NUMBER OF TAKE OFFS
                               AT PROPER INTERVALS
                         *ENDIF
                         *IF (ANCHORS LEFT TO PROCESS FOR THIS AB)
                          THEN
                              *GET ANCHOR POINTER
                              *SET LOOP = 1
                         *END IF
                    *ELSE
                         *SCHEDULE NEXT INTERCEPTOR LAUNCH IN 30
                         SECONDS
                    *ENDIF
               *ENDIF
          *ENDIF
     END DO
*END SEGMENT (BLUTKOF)
```

```
*SEGMENT (COMMAND)
          ●
          ●
          ●
          *IF (INTERCEPTOR OR TANKER ORBIT, ACTION CODE 4) THEN
               *SET STATUS OR ORBITING
          *ENDIF
          ●
          ●
          ●
          *IF (ACTION CODE = 12, FLIGHT ARRIVES AT ANCHOR HEX) THEN
               *INCLUDE (FLTANCR - FLIGHT ARRIVES AT ANCHOR)
          *END IF
          ●
          ●
          ●
*END SEGMENT (COMMAND)
```

```
*SEGMENT (CRCKIL - CRC DEATH RECORD)
     ●
     ●
     ●
     *IF (GOOD GUY) THEN
          ●
          ●
          ●
          *INDICATE APPROPRIATE MESSAGE ON EVENT TRACE
          *IF (TANKER DEATH) THEN
               *SCHEDULE REPLACEMENT TO TAKE OFF AT (TIME + TIME 2AB +
                TABTIME).
          *ELSE
               *DO REST OF GOOD GUY LOOP
               ●
               ●
               ●
          *END IF
     *END IF
          ●
          ●
          ●
*END SEGMENT (CRCKIL)
```

```
*SEGMENT (FUELCHK)
     •
     •
     •
     *IF (NOT ALREADY RETURNING TO AB OR TO TANKER) THEN
          *IF (CAPABLE OF TANKER REFUELING AND AMMO IS REMAINING) THEN
               *INCLUDE (CHZANCR - CHOOSE ANCHOR TO GO TO)
               *IF CNO SUITABLE ANCHORS FOUND) THEN
                    *INCLUDE (GO TO AB - CHANGE OBJECTIVE TO GO TO AB)
               *END IF
          *ELSE
               *INCLUDE (GO TO AB - CHANGE OBJECTIVE TO GO TO AB)
               •
               •
               •
*END SEGMENT (FUELCHK)
```

```
*SEGMENT (CHZANCR - CHOOSE TANKER ANCHOR TO GO TO)
     *IF (THERE ARE ANY ANCHOR AT ALL) THEN
          *GET NUMBER OF HEXES OF FUEL REMAINING
          *FIND (CLOSEST TANKER HEX WITH AVAILABLE FUEL)
          *OF (TANKER HEX IN RANGE) THEN
               *SET UP FLY TO THIS TANKER ORDER
               *SCHEDULE (PONDER - FLIGHT ARRIVES AT ANCHOR)
          *END IF
     *END IF
     *RETURN ("TANKER FOUND" FLAG (OR NOT FOUND)
*END SEGMENT (CHZANCR)
*SEGMENT (FLTANCR - FLIGHT ARRIVES AT ANCHOR HEX)
     *GET IN LINE FIFO QUEUE)
     *SET FLIGHT STATUS TO TANKER ORBIT
     *IF (QUEUE WAS EMPTY) THEN
          *SCHEDULE TANKER PROCESSING
     *END IF
*END SEGMENT (FLTANCR)
```

```
*MODULE TANKER (NEW SELECT MODULE)
*SEGMENT (TANCHOR - TANKER & ANCHOR PROCESSING)
     *IF (TANKERS AVAILABLE) THEN
          *GET NEXT FLIGHT IN QUEUE
          *IF (QUEUE WAS EMPTY) THEN
               *EVALUATE TANKERS' STATUS - CAN SOME BE SENT HOME?
               *IF (TANKERS CAN BE SENT HOME) THEN
                    *SEND THEM TO AB FOR REFUELING
               *END IF
               *SCHEDULE TANKER PROCESSING AT TIME + T OF INT
                (TIME + TAKE OFF INTERVAL)
          *ELSE
               *IF (FLIGHT STILL ALIVE) THEN
                    *DO (UNTIL SCHEDULING COMPLETE)
                         *EVALUATE FLIGHT REFUEL TIME
                           TIME = (TIME PER AC IN FLIGHT) + NO. OF
                                  ACRFT
                           TIME = TIME/NUMBER OF AVAILABLE TANKERS
                         *CHECK TANKERS' FUEL ASSETS FROM TANKERLIST
                          BLOCK:
                           FREETIME = (GAMETIME - TIMEIN) - TIMERLD
                           FUEL = FUELIN-FUELRID - (FREETIME*FUEL
                                  CONSUME)
                         *PUT FUEL VALUE IN ARCFTSTATUS BLOCK
```

TANCHOR -2

```
                    *IF (ANY TANKERS SHORT ON FUEL)
                         *THEN SEND THEM HOME
                    *ELSE
                         *SCHEDULE FLIGHT FINISHED REFUEL
                         *SCHEDULE TANKER-ANCHOR PROCESSING FROM TIME
                          FLIGHT IS FINISHED REFUELING
                         *SET TANKERS BUSY
                    *END IF
               *END PO
          *ELSE
               *SCHEDULE TANKER PROCESSING FOR NOW
          *END IF
     *ELSE
          *EVALUATE QUEUE SIZE AND TANKER RESOURCES
          *IF (QUEUE TOO BIG) THEN
               *SEND SOME FLIGHTS HOME OR TO ANOTHER ANCHOR
          *END IF
     *END IF
*END SEGMENT (TANCHOR)
```

```
*MODULE PONDER
*SEGMENT (FLTFINR - FLIGHT FINISHED REFUELING)
     *IF (FLIGHT STILL ALIVE AND IN TANKER ORBIT STATUS) THEN
          *GIVE FULL FUEL LOAD
          *DO (FOR EACH TANKER USED BY FLIGHT)
               *ADD TO RELOAD TIME AND FUEL
               *COMPLETE FUEL AMOUNTS IN TANKER
               *SET TANKER NOT BUSY
               *IF (TANKER LOW ON FUEL) THEN
                    *SEND IT HOME
               *ELSE
                    *IF (TANKER'S ANCHOR TIME UP AND QUEUE IS EMPTY,
                     AND REPLACEMENT ARRIVED) THEN
                         *SEND IT HOME
                    *END IF
               *END IF
          *END DO
          *SET UP INT CAP AT HIS CRC ORDER - INITIATE CRC ORDER
     *ELSE
          *STUB CODE - FLIGHT LEFT FOR SOME REASON - IGNORE
     *END IF
*END SEGMENT (FLTFINR)
```

```
*SEGMENT (BLULAND - BLUE LANDS AT AIR BASE)
     *IF (FLIGHT STILL ALIVE) THEN
          *WRITE EVENT TRACE - INDICATE INT OR TANKER
          *INCREASE TOTAL AIRCRAFT ON HAND AT AB
          *INCREASE NO. OF AIC OF THAT TYPE ON AB
          *IF (TANKER) THEN
               *SCHEDULE TAKE OFF AT TIME PLUS TABTIME
          *ELSE
               *SCHEDULE INTERCEPTOR LAUNCH AT TIME + 1 HR
          *END IF
          *IF (CRC ALIVE) THEN
               o
               o
               o
*END SEGMENT (BLULAND)
```

2. BVR POL

```
*SEGMENT (TFLYCRC - NORMAL AIRCRAFT FLY)
     •
     •
     •
*IF (EVENT = 1375 = CONSIDER AIR COMBAT) THEN
     *INCLUDE (AIRTANK - NON-BVR ATTACK PONDER)
     *IF (NOT ATTACKING) THEN
          *INCLUDE (BVRTHNK - BVR AIR ATTACK PONDER)
     *END IF
          •
          •
          •
*END SEGMENT (TFLYCRC)
```

```
*SEGMENT (BVRTHNK - BVR AIR ATTACK PONDER)
     *SEGMENT TEMPLATE HEX PATTERN (BASED ON AIRCRAFT TYPE) IN ORDER
      SPECIFIED FOR ENEMY AIRCRAFT.  STOP WHEN FIRST ENEMY AIRCRAFT
      FOUND.  (NOTE:  USE MISID PROBABILITY FOR BUR'S)
     *IF (ENEMY AIRCRAFT FOUND) THEN
          *CALCULATE $P_K$ BASED ON FACTORS DESIRED (I.E., RELATIVE
           SPEED, HEADING, ALTITUDE, AIRCRAFT TYPE, ORDNANCE TYPE,
           TIME OF FLIGHT OF MISSILE)
          *LAUNCH MISSILE AND SCHEDULE ARRIVAL AT TIME = TIME + TOF
           (±t).  THE ±t IS AN OPTION FINE TUNING PARAMETER INPUT BY
          THE USER.
          *MARK FLIGHT IN BVR ATTACKING MODE
     *END IF
*END SEGMENT (BVRTHNK)
```

NEW SELECT MODULE - BVRMAT

```
*SEGMENT (BVRMAT - BVR MISSILE ARRIVES AT TARGET)
     *IF (SHOOTER STILL ALIVE) THEN
          *DRAW RANDOM NUMBER AND COMPARE AGAINST PREVIOUSLY CALCU-
           LATED $P_K$.
          *IF (HIT) THEN
               *DO KILL TGT ACTION
               *SCHEDULE PONDER
          *ELSE
               *DO MISS TGT ACTION
               *SCHEDULE PONDER
          *END IF
          *UNMARK SHOOTER IN BVR ATTACKING MODE
     *ELSE
          *DO SHOOTER DEAD BEFORE MISSILE ARRIVES ACTION
     *END IF
*END SEGMENT
```

3. DISENGAGEMENT PDL

```
*SEGMENT (DOGTHNK - DOGFIGHT PONDOR)
     •
     •
     •
     *RESET FLIGHT CHARACTERISTICS
     *RESET AIR COMBAT INDICATORS
     *IF (NO MUTUAL SUPPORT IN FLIGHT) THEN
          *SET UP CAP AT CRC ORDER
     *END IF
     *EXIST SEGMENT (DOGTHNK)
          •
          •
          •
*END SEGMENT (DOGTHNK)
```

APPENDIX D

PDL FOR MOBILE SAM (LOW MOBILITY) ENHANCEMENT

```
*SEGMENT (RMOBSAM - REALLOCATE MOBILE SAM UNITS)
     SUBROUTINE RMOBSAM
          THIS ROUTINE, EXECUTED, BETWEEN RAIDS, ATTEMPTS TO REALLO-
          CATE MOBILE SAM UNITS TO CREATE A BETTER DEFENSIVE BALANCE
          BEFORE THE NEXT RAID.  THE MOBILE SAMS ARE ORGANIZED INTO
          SUPPLY AND DEMAND GROUPS IN CONJUNCTION WITH PRIORITY LEVELS
          FOR EACH UNIT.  GROUPS WITH THE HIGHEST DEMAND WILL BE FIRST
          TO REPLACE THEIR UNITS.  THE HIGHEST PRIORITY UNITS WILL BE
          REPLACED BY THE NEAREST LOW PRIORITY UNIT IN A SURPLUS
          GROUP.
*COMMONS
*MIDAS DECLARATIONS
*COMPUTE OSR - OVERALL SURVIVAL AVERAGE
*DO (FOR EACH MOBILE SAM GROUP)
     *COMPUTE DEMAND OR SURPLUS FOR GROUP
     *PUT ON DEMAND OR SURPLUS LIST IN SORTED ORDER (HIGH TO LOW FOR
      EACH LIST)
*ENDDO
*DO UNTIL (END OF DEMAND LIST OR SURPLUS LIST)
     *DO (ONCE FOR EACH UNIT DEMANDED)
          *PICK HIGHEST PRIORITY UNIT OF THOSE KNOCKED OUT IN THE
           GROUP AND NOT YET REPLACED.  WE WILL REPLACE THIS UNIT NOW.
          *PICK THE TWO LOWEST PRIORITY AVAILABLE UNITS FROM EACH
           SURPLUS GROUP
          *PICK THE CLOSEST OF THOSE SURPLUS UNITS JUST PICKED.
          *COMPUTE TRAVEL TIME FOR REALLOCATION
          *SCHEDULE ARRIVAL OF SAM AT TIME + TRAVEL TIME
          *TAKE UNIT OFF HEX TREE, UOL, AND PEEPER LISTS
          *SET SURPLUS LOCATION TO INACTIVE
          *SET SAM ITSELF TO INACTIVE
          *UPDATE DEMAND AND SURPLUS LISTS
          *ENDDO
     *ENDDO
*END SEGMENT (RMOBSAM)
```

```
*SEGMENT (AMOBSAM - ARRIVAL OF MOBILE SAM UNIT)
      SUBROUTINE AMOBSAM
            THIS ROUTINE HANDLES AN EVENT - THE ARRIVAL OF A MOBILE SAM
            TO ITS REALLOCATED POSITION.  ALL THAT IS NEEDED FOR THIS
            ROUTINE TO DO IS TO ADJUST THE DATA BASE TO REPRESENT THE
            NEW SAM LOCATION.
*COMMONS
*MIDAS DECLARATIONS
*GET PTR TO SAM SB
*GET PTR TO NEW LOCATION'S MOBILE BLOCK
*SET STATUS TO ACTIVE FOR THIS LOCATION (IN MOBILE BLOCK)
*SET PSB TO POINT TO SAM'S SB
*RESET ADDRESS IN SB
*PUB UNIT ON HEX TREE & ASSOCIATED LISTS
*SET UNIT TO ACTIVE IN STATUS BOARD
*CHANGE POSITION IN C2 TREE (IF NECESSARY)
*END SEGMENT (AMOBSAM)
```

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE

Armed Forces Staff College
ATTN: Reference & Technical Svcs Br

Assistant Secretary of Defense
International Security Affairs
ATTN: European & NATO Affairs
ATTN: Policy Plans & NSC Affairs

Assistant Secretary of Defense
Comm, Cmd, Cont & Intell
ATTN: Surveillance & Warning Sys

Assistant to the Secretary of Defense
Atomic Energy
ATTN: Executive Assistant

Commander-in-Chief, Atlantic
ATTN: N-22

Commander-in-Chief, Pacific
ATTN: J-634
ATTN: J-54
ATTN: J-32

Defense Intelligence Agency
ATTN: DIA/VPA-2, Fed Res Div
ATTN: DB-6
5 cy ATTN: DB-4

Defense Nuclear Agency
ATTN: STNA
ATTN: NATD
ATTN: NATA
4 cy ATTN: TITL

Defense Technical Information Center
12 cy ATTN: DD

Field Command
Defense Nuclear Agency
ATTN: FCPR

Field Command
Defense Nuclear Agency
Livermore Branch
ATTN: FCPRL

Intelligence Center, Pacific
ATTN: COMIPAC

Interservice Nuclear Weapons School
ATTN: TTV

Joint Chiefs of Staff
ATTN: J-3
ATTN: J-5, Strategy Division, W. McClain
ATTN: J-5, Nuclear Division
ATTN: J-3, Strategic Operations Div
ATTN: SAGA/SSD
ATTN: SAGA/SFD

National Defense University
ATTN: NWCLB-CR
ATTN: ICAF, Tech Lib

DEPARTMENT OF DEFENSE (Continued)

Joint Strat Tgt Planning Staff
ATTN: JL
ATTN: JLA
ATTN: JP
2 cy ATTN: JLTW-2

Director
NET Assessment
Office of the Secretary of Defense
ATTN: Military Assistants

U.S. European Command
ATTN: J-2
ATTN: J-3
ATTN: J-LW
ATTN: J-5NPG
ATTN: J-6
ATTN: J-2-ITD

U.S. National Military Representative
SHAPE
ATTN: U.S. Documents Officer

Under Secretary of Defense for Rsch & Engrg
ATTN: Strategic & Space Sys (OS)
ATTN: Tactical Warfare Programs

DEPARTMENT OF THE ARMY

Deputy Chief of Staff for Ops & Plans
Department of the Army
ATTN: DAMO-ZXA
ATTN: DAMO-NC

Harry Diamond Laboratories
Department of the Army
ATTN: DELHD-N-P

U.S. Army Comd & General Staff College
ATTN: ACQ, Library Div

U.S. Army Concepts Analysis Agency
ATTN: CSSA-ADL

Commander-in-Chief
U.S. Army Europe and Seventh Army
ATTN: AEAGC
ATTN: AEAGE

U.S. Army Nuclear & Chemical Agency
ATTN: Library

U.S. Army TRADOC Sys Analysis Actvy
ATTN: ATAA-TAC

U.S. Army War College
ATTN: Library

U.S. Military Academy
Department of the Army
ATTN: Document Library

DEPARTMENT OF THE NAVY

Naval Postgraduate School
ATTN: Code 1424, Library

Naval Research Laboratory
ATTN: Code 1240

Naval Sea Systems Command
ATTN: SEA-09G53
ATTN: SEA-643

Naval Surface Weapons Center
ATTN: Code F31

Naval War College
ATTN: Document Control

Nuclear Weapons Tng Group, Pacific, Dept of Navy
ATTN: Document Control

Nuclear Weapons Tng Group, Atlantic, Dept of Navy
ATTN: Document Control

Office of Naval Research
ATTN: Technical Info Services

DEPARTMENT OF THE AIR FORCE

Air Force Systems Command
ATTN: DL
ATTN: XR
ATTN: SD

Air Force Weapons Laboratory
Air Force Systems Command
ATTN: SUL

Air University Library
Department of the Air Force
ATTN: AUL-LSE

Assistant Chief of Staff
Intelligence
Department of the Air Force
ATTN: INA

Deputy Chief of Staff
Research, Development, & Acq
Department of the Air Force
ATTN: AFRDQI

Foreign Technology Division
Air Force Systems Command
ATTN: CCN
ATTN: TQTM
ATTN: SDN

Strategic Air Command
Department of the Air Force
ATTN: NR
ATTN: ADWN
ATTN: XP
ATTN: NRI, STINFO Library
ATTN: DO

U.S. Air Force, Pacific
ATTN: XP

Commander-in-Chief
U.S. Air Forces in Europe
ATTN: USAFE/DEX

DEPARTMENT OF THE AIR FORCE (Continued)

Commander-in-Chief
U.S. Air Forces in Europe
ATTN: USAFE/DOT

Commander-in-Chief
U.S. Air Force in Europe
ATTN: USAFE/INA

Commander-in-Chief
U.S. Air Forces in Europe
ATTN: USAFE/INT

Commander-in-Chief
U.S. Air Forces in Europe
ATTN: USAFE/XPX

USAF Tactical Fighter Weapons Center
ATTN: Commander

DEPARTMENT OF ENERGY

Department of Energy
Albuquerque Operations Office
ATTN: CTID

OTHER GOVERNMENT AGENCY

Central Intelligence Agency
ATTN: OSWR/NED

DEPARTMENT OF ENERGY CONTRACTORS

Lawrence Livermore National Lab
ATTN: Tech Info Dept, Library

Los Alamos National Lab
ATTN: MS 364
ATTN: M/S634, T. Dowler

DEPARTMENT OF DEFENSE CONTRACTORS

Advanced International Studies Institute
ATTN: M. Harvey

Aerospace Corp
ATTN: Library

BDM Corp
ATTN: J. Braddock
4 cy ATTN: M. Filteau
4 cy ATTN: T. Hawkins
4 cy ATTN: L. Elfes
4 cy ATTN: B. Macaleer
4 cy ATTN: J. Phillips

JAYCOR
ATTN: R. Sullivan

Kaman Sciences Corp
ATTN: F. Shelton

Kaman Sciences Corp
ATTN: E. Daugs

Kaman Tempo
ATTN: DASIAC

Pacific-Sierra Research Corp
ATTN: G. Lang
ATTN: H. Brode

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Pacific-Sierra Research Corp
ATTN: D. Gormley

R & D Associates
ATTN: R. Port
ATTN: P. Haas
2 cy ATTN: Document Control
2 cy ATTN: S. Borjon

R & D Associates
ATTN: J. Thompson

Science Applications, Inc
ATTN: Document Control
ATTN: J. Warner
ATTN: M. Drake

Science Applications, Inc
ATTN: W. Layson
ATTN: Document Control

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Science Applications, Inc
ATTN: D. Kaul
ATTN: L. Goure

System Planning Corp
ATTN: J. Douglas

Tetra Tech, Inc
ATTN: F. Bothwell

TRW Defense & Space Sys Group
ATTN: N. Lipner

TRW Defense & Space Sys Group
ATTN P. Dai